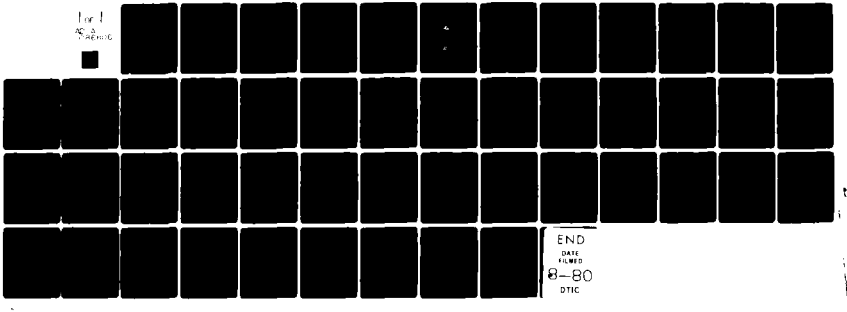
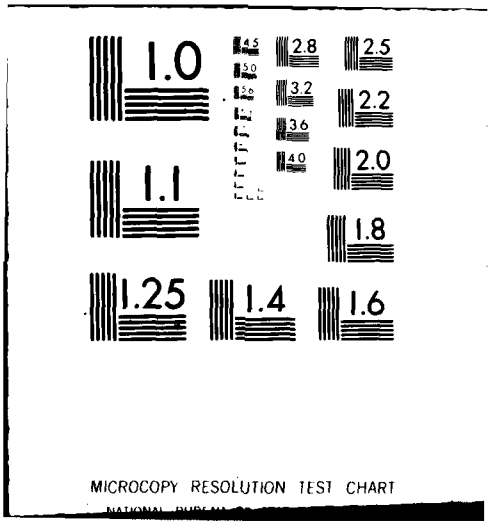
AD-A086 606 ORINCON CORP LA JOLLA CA F/G 9/2
DATA ASSOCIATION ALGORITHMS FOR LARGE AREA SURVEILLANCE,(U)
1978 C L MOREFIELD, C M PETERSEN N00014-77-C-0296
UNCLASSIFIED NL



END
DATE FILMED
8-80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART

DOCUMENT 5

DATA ASSOCIATION ALGORITHMS FOR LARGE AREA SURVEILLANCE

ORINCON

This work was supported by the Office of Naval Research
under Contract N00014-77-C-0296.
(15)

DATA ASSOCIATION ALGORITHMS FOR LARGE AREA SURVEILLANCE

ORSA/TIMS Meeting

May, 1978

(10) Dr. Charles L. Morefield*
and
Christian M. Petersen**

(11) 1978
(12) 149

392776

*1745 Collingwood Drive, San Diego, CA 92109
**8571 Villa La Jolla, #D, La Jolla, CA 92037

392776

1.0 OVERVIEW OF INTERSENSOR CORRELATION FOR OCEAN SURVEILLANCE

In a typical large-area ocean surveillance situation, data is generated by many different sensors due to the presence of various surface ships, aircraft, etc. If the data collected by the sensors is overlaid in a common coordinate system, then a picture such as that shown in Figure 1-1 below results. The underlying assumption of the picture is that each individual sensor has formed a picture of the surveillance area based only on its own data. In some cases, this may require the solution of a multitarget tracking problem [1] at the sensor level. In Figure 1-1, for example, sensor S2 must discriminate and track the two closely spaced targets T1 and T2.

When the multitarget tracking problem is solved separately for each individual sensor, a somewhat redundant view of the surveillance area may result. The different data collection systems S1, S2, ..., may report on the same target. When this situation arises, the final step in the process of forming a complete picture of the surveillance area is to perform intersensor correlation. Referring again to Figure 1-1, we see that target T1 is seen by sensors S1 and S2. The tracks of T1 produced by the two sensor systems will not overlay exactly--thus the requirement for some sort of decision process (automatic or manual) to accomplish intersensor correlation.

This paper is devoted exclusively to automatic (i.e., computer) algorithms for carrying out the tasks associated with intersensor correlation. One particular algorithm (the "k-track clustering" algorithm) is discussed in some detail in order to highlight the fundamental combinatorial structure of large area surveillance.

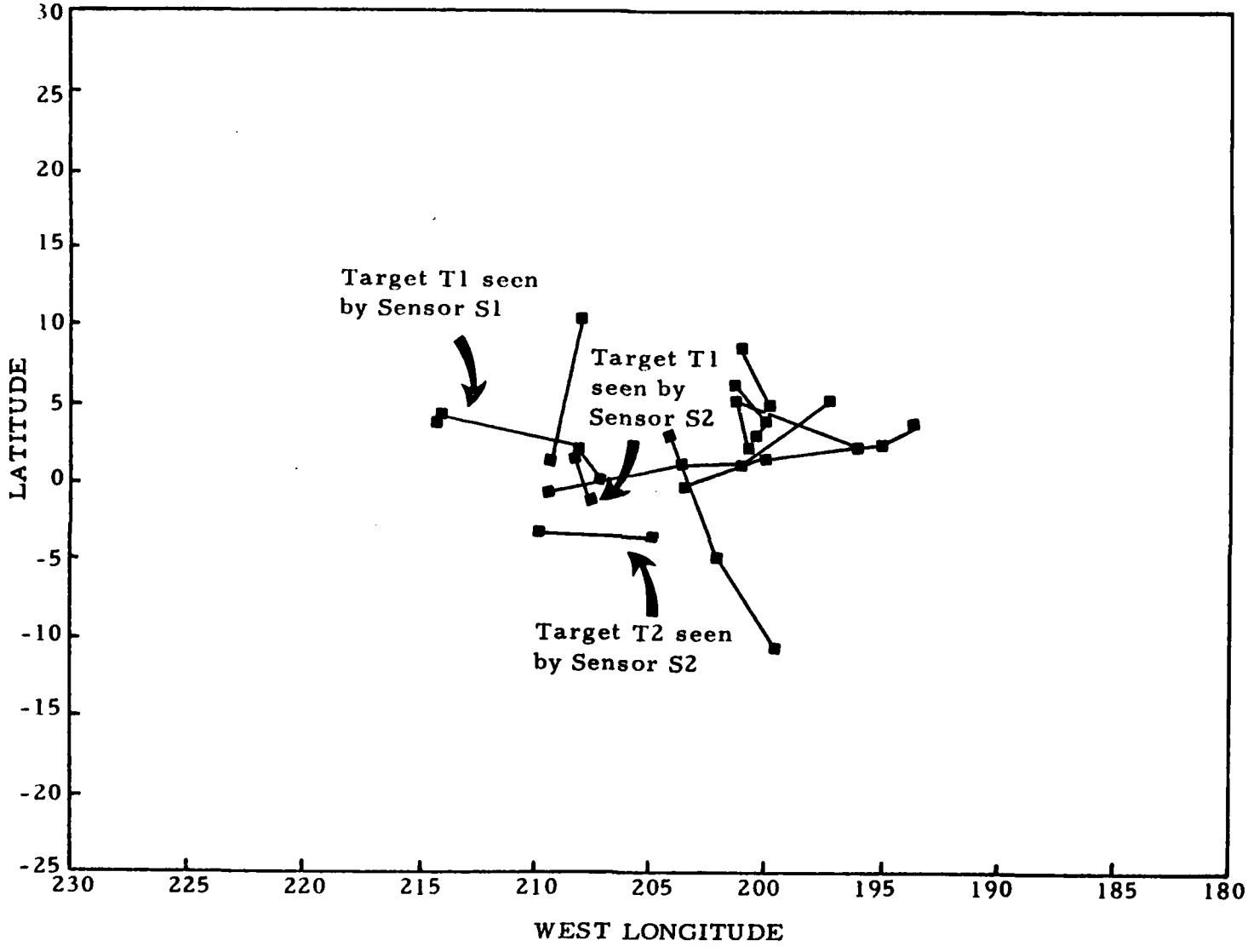


Figure 1-1. Typical multisensor view of a large ocean area.

The working definition of Multisource Interaction (MSI) adopted in this paper is one which is effectively decoupled from the problems of sensor allocation and sensor-level multitarget tracking. By that we mean that the sensors' fields of view are controlled by a separate process, and that the multitarget tracking problem is solved on a sensor-by-sensor basis. The responsibility of MSI data processing is then to carry out intersensor correlation for various track file data bases. The MSI issues addressed herein arise due to the partitioning of a large surveillance data base into smaller, more manageable segments. As Figure 1-2 indicates, data is generated by many sensors due to objects present in several spatial sectors during some specified time interval. If the multitarget tracking problem is solved separately for each individual sensor (see [1]), then track estimates for objects in the surveillance area become consolidated first at the level of individual sensors. This is indicated schematically in Figure 1-2 as the consolidation of the raw sensor measurements (data bases A, B, C) into a "sensor-level" data base D that consists of a complete track file for that specific sensor.

The final step in the process of forming a complete picture of the objects present in the surveillance area is the solution to the intersensor correlation problem. At this level of the MSI data processing hierarchy, the track files D, E, F belonging to the different sensor systems are compared to determine the correlation between tracks in separate data bases. As indicated in Figure 1-2, the output of an intersensor correlation algorithm is an entry (or entries) into a master data base G that consists of track files on an object-by-object basis for the entire surveillance volume. If the intersensor correlation problem is solved correctly, the master data base will contain a <u>complete list</u> of object tracks, <u>without duplication</u>, and <u>in</u>

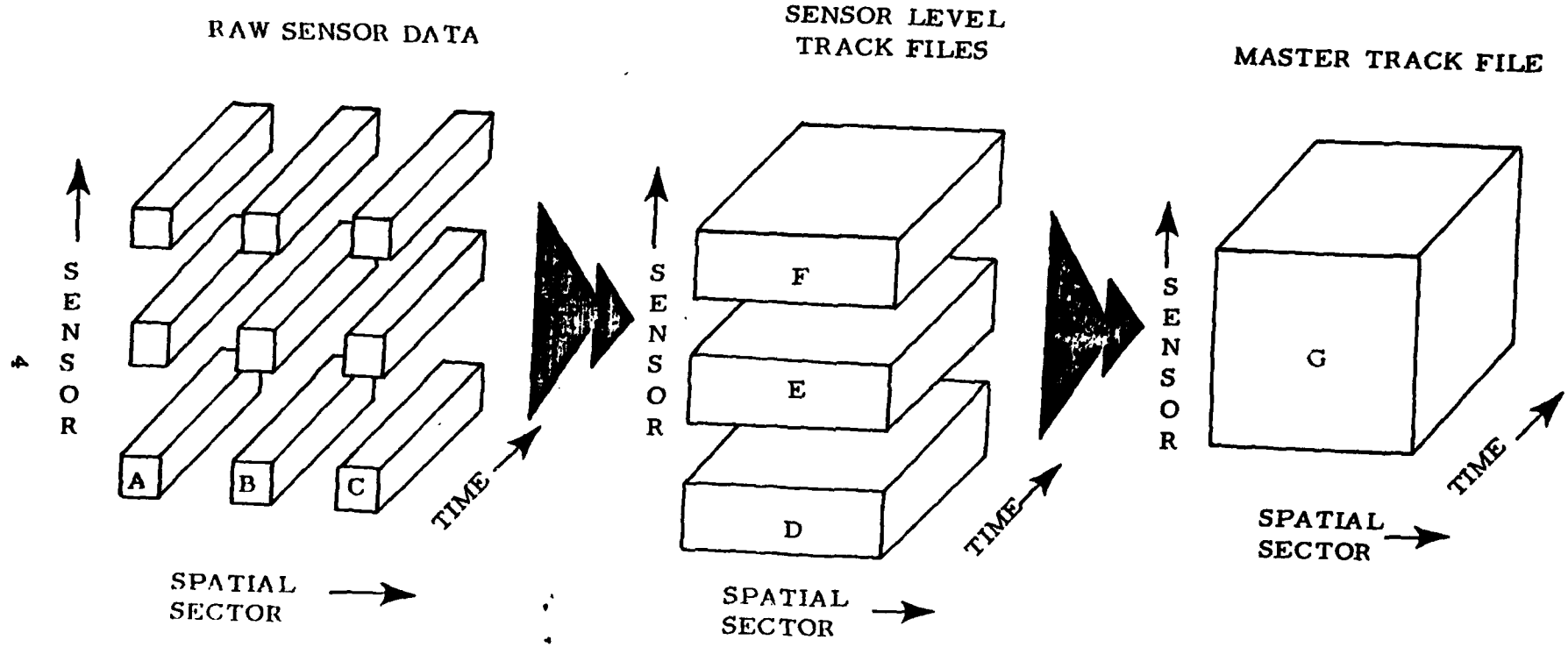


Figure 1-2. Overview of MSI Hierarchical Information Processing.

a single, common coordinate system. (The coordinate system utilized in the algorithm described in Section 2. is latitude, latitude rate, longitude, longitude rate.)

1.1 Description of the Intersensor Correlation Problem

The multitarget tracking data processing done at the sensor level, using algorithms such as those discussed in [1], produces track files consisting of object state vectors and (sometimes) estimation error covariance at specific points in time.

Typically, the sensor level track files will contain information such as that shown below in Table 1-1.

Table 1-1. Typical information content of an *MSI Data Base*.

Data / Track	State Vector	State Vector Error Covariance	Object Name	Sensor I.D. Number
1	$\hat{x}^{3,1}(t_1)$, $\hat{x}^{3,1}(t_2)$	$P^{3,1}(t_1)$, $P^{3,1}(t_2)$	QUEEN MARY	3
2	$\hat{x}^{8,2}(t_3)$	Unknown	Unknown	8
3	$\hat{x}^{6,3}(t_4)$, $\hat{x}^{6,3}(t_5)$, $\hat{x}^{6,3}(t_6)$	$P^{6,3}(t_4)$, $P^{6,3}(t_5)$, $P^{6,3}(t_6)$	Unknown	6

The notation used in Table 1-1 is defined as follows:

$\hat{\underset{\sim}{x}}^{j,i}(t_k)$ ~ sensor level track file estimate of the state at time t_k of a target seen by sensor j. The individual tracks are denoted by the index i.

$P^{j,i}(t_k)$ ~ the sensor level track file error covariance matrix of the measurement $\hat{\underset{\sim}{x}}^{j,i}(t_k)$.

It is often the case that the $(\hat{\underset{\sim}{x}}, P)$ data recorded in the data base is itself generated by a tracking filter of some kind. Referring again to Figure 1-2, recall that the $(\hat{\underset{\sim}{x}}, P)$ records are associated with sensor level track files (D, E, F). Thus, $(\hat{\underset{\sim}{x}}, P)$ records are created by processing the raw sensor data for <u>one</u> sensor with a filter to produce a chain of target track points.

Because of the diversity of sensors that are operating in a large area surveillance system, the time points t_k at which the object states (latitude, longitude, etc.) are estimated may not coincide. Furthermore, the number of time points per object and even the state vector coordinate system where the sensor-level multitarget tracking problem is resolved will differ for the various sensors' track files. The diverse nature of the sensor-level track files introduces an added degree of complexity into the intersensor correlation problem.

Based on these last comments, a more accurate definition of intersensor correlation might be as follows: to cluster together those tracks corresponding to the same object, and to produce a composite estimate of each object's motion in a single coordinate system common to all sensors.

1.2 Algorithms for Intersensor Correlation

There are two basic components of any intersensor correlation process: a similarity measure to quantify the "distance" between tracks, and a clustering criterion to carry out the actual correlation of tracks. Since the tracks to be clustered together are in fact samples from a stochastic process, similarity measures can be based theoretically upon the distance between stochastic processes. In essence, we simply process the data from different track files with a single Kalman filter; the residuals from the filter are then used to measure the "distance" between the tracks.

The clustering criteria used to determine the intersensor correlations can range from various forms of pairwise correlators to the more sophisticated k-wise correlation to be discussed below. As an example of a simple pairwise clustering criterion, consider the following:

Step 1. Compute the similarity S_{ij} between all pairs of tracks

Step 2. If S_{ij} is greater than a preset threshold δ, then Track i and Track j are declared to be the same object ($i \leftrightarrow j$)

Step 3. If $i \leftrightarrow j$, and $j \leftrightarrow k$, then $i \leftrightarrow k$ (i.e., all three tracks are the same ship).

A little reflection upon the algorithm presented above will indicate to the reader that an iterative application of Step 3 will produce a complete picture of the correlations in any particular data base. It is also clear that the algorithm is vulnerable to at least one type of cascading error. This is shown below in Figure 1-3, which contains four targets (T1, T2, T3, T4) on parallel courses. If S_{12}, S_{23}, S_{34} are all greater than δ, then the pairwise algorithm outlined above will declare all four sensor-level track files to be the same ship.

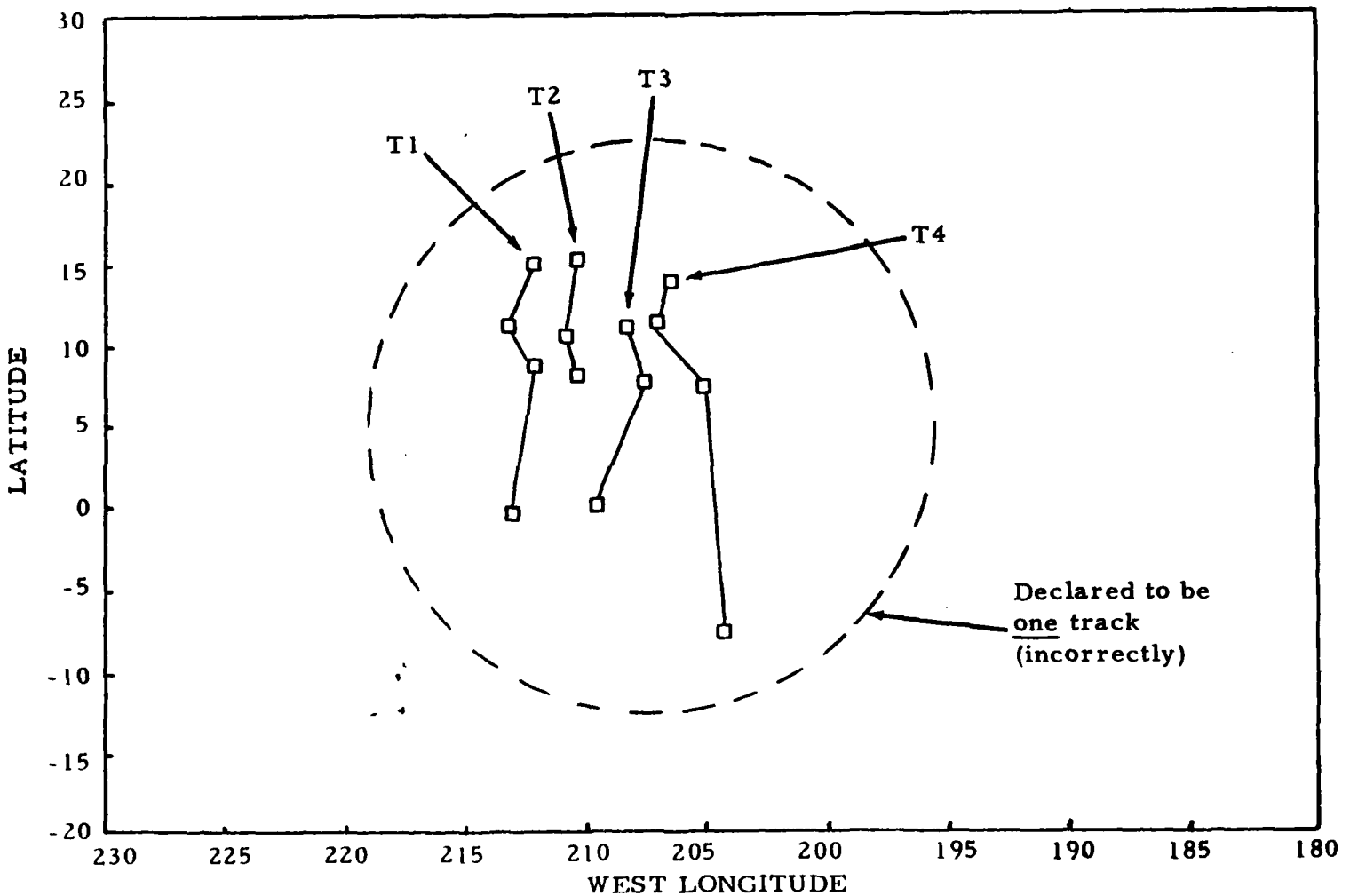


Figure 1-3. Cascading errors in a simple pairwise intersensor correlation process.

The object of these last remarks is to illustrate that simple automatic algorithms for intersensor correlation may have hidden difficulties, and that there is some merit in working with the more complex decision processes discussed in the following sections.

1.2.1 Track File Similarity Measure

This section will describe a simple technique for measuring the similarity between different tracks in an MSI data base. For example, in Table 1-1 above, a fundamental question that the analyst might wish to ask is the following: how similar is track #1 (QUEEN MARY) to track #3? The basic technique we employ is illustrated in Figure 1-4 below. First, the points t_1, t_2, t_3, t_4, t_5, t_6 are ordered in terms of increasing time. Following this, the sensor-level state vectors are input to a Kalman filter, and the residual sequence is monitored.

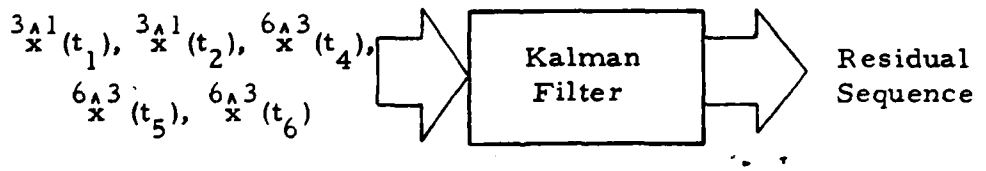


Figure 1-4. Measuring the similarity of different track files.

When testing the similarity of two tracks i and j, it is necessary that the test be carried out in a coordinate system common to both tracks. Since the tracks i and j come from different sensor-level track files, it is possible that they are represented as vectors $\hat{x}$ of different form. As a simple example, the state of track i might be represented as latitude/longitude only, while track j might be represented as latitude, longitude, course, and speed.

In the algorithm described in this paper, sensor-to-sensor track correlation is carried out for all tracks in a single, common coordinate system. We assume that target motion is adequately modeled by a stochastic difference equation of the form

$$y(k+1) = \varphi(k+1, k)\, y(k) + v(k), \quad k = 0, 1, \ldots, \tag{1}$$

where

- $y(k)$ ~ vector describing object state at time t_k (latitude, latitude rate, longitude, longitude rate)
- $\varphi(\cdot, \cdot)$ ~ state transition matrix model (for "straight-line" motion)
- $v(k)$ ~ noise model compensating for inaccuracies in modeled track motion (e.g., small random course changes)

The state vectors $\hat{x}$ from the track files are treated as if they are "measurements" of actual target motion; the error covariance matrices P of the state vectors are treated as though they are "noise" in the measurements. Using this pseudo-measurement, we have:

$$^{j}\hat{x}^{i}(k) = c^{j}(y(k), k) + {}^{j}w^{i}(k) \tag{2}$$

where

- $y(k)$ ~ actual target state at time t_k
- $c^{j}(\cdot, \cdot)$ ~ observation equation for j^{th} sensor
- $^{j}w^{i}(k)$ ~ "measurement" noise with covariance $^{j}P^{i}(t_k)$

Using the $\hat{x}$ as inputs, a Kalman filter can be used to estimate the true ship motion y(k) under the assumption that the inputs $\hat{x}$ from the various sensor-level track files are due to the same ship. As the Kalman filter operates, residuals r(k) are produced whose magnitude can be predicted, using the covariances P from the track files and the covariances of v(k) in Equation (1) above. In particular, the covariance matrix $\Delta(k)$ of the residuals can be computed.

If the observed variance of the residuals r(k) compares favorably with their predicted value $\Delta(k)$, then the sensor-level tracks whose state vectors were used to compute the residual sequence $\{r(k)\}$ are said to be "similar."

Note that in the two-track case, if the vectors for the two tracks i and j being tested for correlation are in the same coordinate system and at the same time point, then we could alternatively use euclidean distance as a similarity measure, for example

$$\text{distance } (i,j) = \| {}^{k}\hat{x}^{i}(t) - {}^{\ell}\hat{x}^{j}(t) \|^{2} \tag{3}$$

$$S_{ij} = \exp(-\text{distance } (i,j)) \tag{4}$$

However, the tracks we deal with from the various MSI files are often not in the same coordinate system and not at the same time point. Therefore, we resort to the Kalman filtering technique described above to compute distance between tracks.

Each potential correlation ω is the result of a hypothesis test that uses a likelihood function $p(\omega)$ determined from the motion model Equation (1). For example, if Track #1 and Track #3 from Table 1-1 are to be tested to determine their similarity, then

$$\omega = \left\{ {}^{3}\hat{x}^{1}(t_1),\ {}^{3}\hat{x}^{1}(t_2),\ {}^{6}\hat{x}^{3}(t_4),\ {}^{6}\hat{x}^{3}(t_5),\ {}^{6}\hat{x}^{3}(t_6) \right\}$$

is a 2-track correlation and the likelihood function is found by computing a sequence of estimates $\hat{y}(t_1)$ of ship position using Equations (1) and (2) and ω. The rule that decides that tracks $i_1, i_2, i_3, \ldots$ are correlated is then

$$S_{i_1 i_2 i_3 \ldots} = \ell n\, p\left(\omega \mid \{\hat{y}(t_k)\}\right) \geq \delta . \tag{5}$$

Based on the log likelihood decision function, the set of potential correlations is

$$D = \left\{ \omega \mid \ell n\, p\left(\omega \mid \{\hat{y}(t_k)\}\right) \geq \delta \right\} . \tag{6}$$

As noted above, the actual computation of $S_{i_1 i_2 i_3 \ldots}$ is carried out by a Kalman filter, using the innovations sequence

$$r(k) = {}^{j}\hat{x}^{i}(k) - c^{j}(\varphi(k, k-1)\hat{y}(k-1), k) . \tag{7}$$

The negative log-likelihood function is given in this case by

$$-\ell n\, p(\omega) = \frac{1}{2} \sum_{k=1}^{n} \dim\left({}^{j}\hat{x}^{i}(t_k)\right) \ell n\, 2\pi + \frac{1}{2} \sum_{k=1}^{n} \ell n\, |\Delta(k)| .$$

$$+ \frac{1}{2} \sum_{k=1}^{n} r(k)^{t} \Delta(k)^{-1} r(k), \tag{8}$$

where $\Delta(k)$ is the covariance of $r(k)$ computed by the Kalman filter and $\dim({}^{j}\hat{x}^{i}(t_k))$ is the dimension of ${}^{j}\hat{x}^{i}(t_k)$.

The summation $\Sigma r(k)^t \Delta(k)^{-1} r(k)$ is a chi-squared random variable, as are its individual terms $r(k)^t \Delta(k)^{-1} r(k)$. The remaining terms in Equation (8) are deterministic. As a result, the similarity test Equation (5) can be written in the form of the following cumulative chi-square test:

$$\sum_{k=1}^{n} r(k)^t \Delta(k)^{-1} r(k) \leq \chi_c^2 . \qquad (9)$$

Note that χ_c^2 is a function of the number of individual measurements in the correlation.

The actual construction of the complete set of potential correlations D is accomplished through the use of a depth first backtracking algorithm. By this we mean that as many tracks as possible are added to a potential correlation before backtracking occurs. When backtracking occurs, all points associated with the track being stripped out of the correlation are removed.

1.2.2 Discriminating Between Correct and Incorrect Correlations

Because of the statistical nature of the decision process (9) used to construct the potential correlation file D, some tracks may be included in more than one correlation. It may be difficult to decide which correlation is correct (if any) based only on the similarity measure. The Bayesian decision process is structured to alleviate this difficulty by evaluating only complete pictures of the surveillance area. The potential correlations in D are matched together in every way possible until the best possible "global" picture of the area is found. The basic constraint operable at each stage in this combinatorial decision problem is that a track can be used in at

most one correlation. The actual optimization problem is the following:

maximize $\Sigma S_{i_1 i_2 i_3 \cdots}$ subject to the constraint that tracks are only used once

(10)

The mathematical form of the problem is as follows:

$$\max \; d^t \nu \tag{11}$$

subject to

$$B\nu \leq \mathbf{1} \tag{12}$$

$$\nu \text{ binary} \tag{13}$$

where d is a vector of similarity measures $S_{i_1 i_2 i_3 \ldots}$, ν is a binary vector denoting which elements of D were selected, B is a matrix of zeros and ones chosen to enforce the constraint mentioned in (10), and $\mathbf{1}$ is a vector of ones.

1.2.3 Integer Programming Methods

Problem (11) – (13) is a 0-1 integer program that can be solved using implicit enumeration techniques. The basic idea of these techniques is to minimize through the use of appropriate tests the extent to which the Bayesian decision tree must be examined before a solution of (11) – (13) is found. The actual technique employed in the algorithm is discussed in some detail in [1]. Since the combinatorial structure of the Bayesian decision process is the same for both k-track correlation and the multitarget tracker discussed in [1], the algorithm can be used without modification.

2.0 NUMERICAL RESULTS

This chapter will discuss some of the characteristics of the k-track correlation process as it is actually applied to data. Table 2-1 below summarizes a set of two test cases that were analyzed. Runs #1 and 2 were synthetic data cases in which intermittent, noisy track reports were made on six ships. The synthetic data test cases will be discussed in some detail to illustrate the basic structure of the MSI correlation problem.

Table 2-1. Summary of numerical results using k-track correlation algorithm.

Run No.	Data File	Parameter File	Number of Potential Correlations	Time Required for Potential Correlations (sec.)	Number of Correlations Selected	Time Required by Integer Program (sec.)	Number of Individual Track Reports in Data Base	Number of Data Points
1	ORT1A	PAR1A	11	5.7	3	<.1	10	30
2	ORT1B	PAR1B	26	7.4	3	.3	10	30

2.1 Description of Simulated Data Bases

The truth model for Runs #1 and 2 is contained in Figure 2-1 and Tables 2-2 and 2-3 below. The actual data bases (ORT1A and ORT1B) used in Runs #1 and 2 are contained in Tables 2-4 and 2-5.

The situation modeled in data bases ORT1A and ORT1B is one in which six simulated ships are seen and reported by a fictitious suite of MSI sensors on different occasions within a 60-hour time period. The ships are assumed to hold constant latitude rate/longitude rate courses for the entire 60 hours. The sensors that are brought to bear on the situation are of two types, as indicated in Table 2-6. Type 1 sensors measure latitude, longitude, and their time derivatives. Type 2 sensors measure latitude and longitude only.

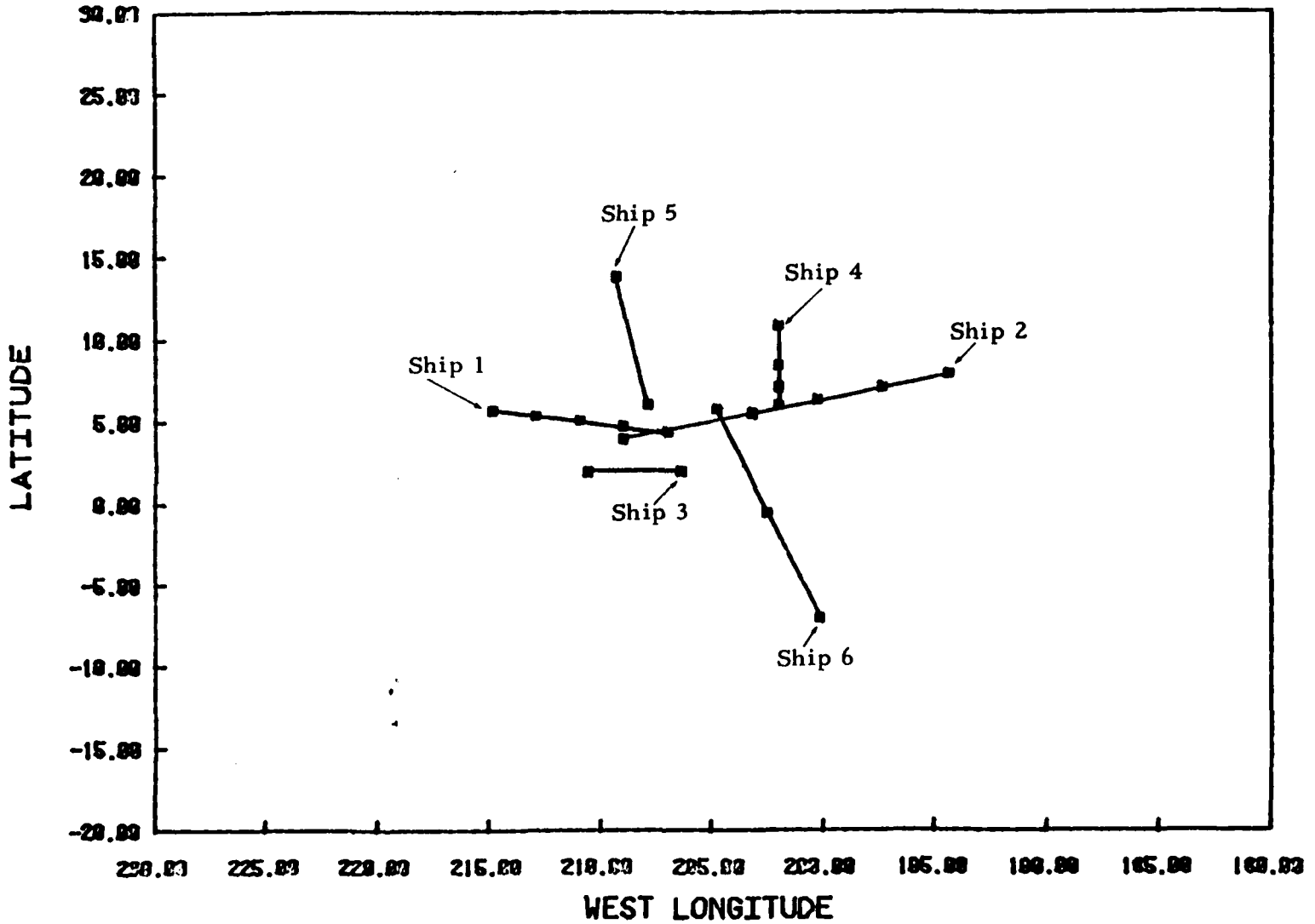


Figure 2-1. Six-ship simulation scenario (truth model).

Table 2-2. Summary of 6-ship simulation.

Ship	Initial N. Lat. (deg.)	Initial E. Long. (deg.)	Course (deg.)	Speed (knots)	Number of Sensors Detecting Ship and Forming Track Report
1	4	155	280	10	2
2	4	151	75	15	3
3	2	155	270	7	1
4	6	158	0	6	2
5	1	153	350	13	1
6	9	154	160	17	1

Table 2-3. Truth model for Runs #1 and 2.

DATA POINT NUMBER	SHIP NUMBER (TRUTH)	TRACK FILE NUMBER	MOVEMENT INFORMATION				
			TIME	NORTH LAT	LATITUDE RATE	EAST LONG	LONGITUDE RATE
1	1	1	43200.0	4.3473	0.8039E-05	153.0251	-0.4572E-04
2		1	129600.0	5.0419	0.8039E-05	149.0726	-0.4577E-04
3		2	43200.0	4.3473	0.8039E-05	153.0251	-0.4572E-04
4		2	86400.0	4.6946	0.8039E-05	151.0494	-0.4575E-04
5		2	172800.0	5.3892	0.8039E-05	147.0948	-0.4580E-04
6		2	216000.0	5.7365	0.8039E-05	145.1159	-0.4582E-04
7	2	3	86400.0	5.5529	0.0000E+00	156.8159	0.0000E+00
8		3	129600.0	6.3294	0.0000E+00	159.7294	0.0000E+00
9		3	172800.0	7.1058	0.0000E+00	162.6472	0.0000E+00
10		4	0.0	4.0000	0.1797E-04	151.0000	0.6724E-04
11		4	86400.0	5.5529	0.1797E-04	156.8159	0.6739E-04
12		4	129600.0	6.3294	0.1797E-04	159.7294	0.6749E-04
13		4	172800.0	7.1058	0.1797E-04	162.6472	0.6760E-04
14		5	86400.0	5.5529	0.1797E-04	156.8159	0.6739E-04
15		5	129600.0	6.3294	0.1797E-04	159.7294	0.6749E-04
16		5	172800.0	7.1058	0.1797E-04	162.6472	0.6760E-04
17		5	216000.0	7.8823	0.1797E-04	165.5700	0.6772E-04
18	3	6	43200.0	2.0000	0.0000E+00	153.5991	0.0000E+00
19		6	172800.0	2.0000	0.0000E+00	149.3966	0.0000E+00
20	4	7	0.0	6.0000	0.2778E-04	158.0000	0.0000E+00
21		7	172800.0	10.8000	0.2778E-04	159.0000	0.0000E+00
22		8	0.0	6.0000	0.2778E-04	158.0000	0.0000E+00
23		8	43200.0	7.2000	0.2778E-04	158.0000	0.0000E+00
24		8	86400.0	8.4000	0.2778E-04	158.0000	0.0000E+00
25		8	172800.0	10.8000	0.2778E-04	158.0000	0.0000E+00
26	5	9	86400.0	6.1210	0.5927E-04	152.0950	-0.1051E-04
27		9	216000.0	13.8025	0.5927E-04	150.7187	-0.1076E-04
28	6	10	43200.0	5.8050	-0.7396E-04	155.1728	0.2706E-04
29		10	129600.0	-0.5849	-0.7396E-04	157.5022	0.2692E-04
30		10	216000.0	6.9740	-0.7396E-04	159.8342	0.2712E-04

Table 2-4. Synthetic MSI data base for Run #1 (data base ORT1A).

	MOVEMENT INFORMATION				
	TIME	NORTH LAT	LATITUDE RATE	EAST LONG	LONGITUDE RATE
1	43200.0	4.1492	0.9206E-05	152.9049	-0.4658E-04
2	129600.0	5.0456	0.8344E-05	149.2870	-0.4341E-04
3	43200.0	4.2937	0.5911E-05	152.9590	-0.4638E-04
4	86400.0	4.8076	0.8067E-05	151.0732	-0.4574E-04
5	172800.0	5.5932	0.9221E-05	146.9504	-0.4462E-04
6	216000.0	5.8718	0.5875E-05	145.1482	-0.4542E-04
7	86400.0	5.5847	0.0000E+00	156.9793	0.0000E+00
8	129600.0	6.5455	0.0000E+00	159.5486	0.0000E+00
9	172800.0	6.9731	0.0000E+00	162.6776	0.0000E+00
10	0.0	3.8903	0.1803E-04	150.9037	0.6873E-04
11	86400.0	5.4676	0.1771E-04	156.6906	0.6771E-04
12	129600.0	6.1707	0.1763E-04	159.5567	0.6677E-04
13	172800.0	7.2429	0.1972E-04	162.5593	0.6782E-04
14	86400.0	5.3501	0.1733E-04	156.6992	0.6804E-04
15	129600.0	6.2073	0.1607E-04	159.6627	0.6680E-04
16	172800.0	6.9876	0.1758E-04	162.7621	0.6848E-04
17	216000.0	7.7721	0.1728E-04	165.5123	0.6816E-04
18	43200.0	1.8336	0.0000E+00	153.6828	0.0000E+00
19	172800.0	1.8522	0.0000E+00	149.4227	0.0000E+00
20	0.0	6.1539	0.2936E-04	158.1071	-0.6317E-06
21	172800.0	10.6991	0.2757E-04	157.9753	0.1652E-05
22	0.0	6.0474	0.2727E-04	158.0660	-0.4371E-06
23	43200.0	7.2062	0.2697E-04	158.1080	-0.3568E-05
24	86400.0	8.3819	0.2803E-04	158.1236	0.6289E-06
25	172800.0	10.8695	0.2667E-04	158.0033	-0.9170E-06
26	86400.0	5.9870	0.5672E-04	151.9055	-0.1170E-04
27	216000.0	13.7665	0.6102E-04	150.7840	-0.1249E-04
28	43200.0	5.8855	-0.7186E-04	155.1613	0.2583E-04
29	129600.0	-0.6172	-0.7366E-04	157.4634	0.2694E-04
30	216000.0	-6.9056	-0.7588E-04	159.7972	0.2731E-04

Table 2-5. Synthetic data base for Run #2 (data base ORT1B).

	X----------MOVEMENT INFORMATION----------X				
	TIME	NORTH LAT	LATITUDE RATE	EAST LONG	LONGITUDE RATE
1	43200.0	2.3662	0.1931E-04	151.8230	-0.5395E-04
2	129600.0	5.0791	0.1099E-04	151.2165	-0.2292E-04
3	43200.0	3.8109	-0.1254E-04	152.3634	-0.5202E-04
4	86400.0	5.8242	0.8303E-05	151.2872	-0.4569E-04
5	172800.0	7.4293	0.1946E-04	145.6509	-0.3447E-04
6	216000.0	7.0897	-0.1288E-04	145.4386	-0.4188E-04
7	86400.0	5.8711	0.0000E+00	158.4497	0.0000E+00
8	129600.0	8.4906	0.0000E+00	157.9215	0.0000E+00
9	172800.0	5.7781	0.0000E+00	162.9507	0.0000E+00
10	0.0	2.9031	0.1850E-04	150.0369	0.8164E-04
11	86400.0	4.6994	0.1543E-04	155.5623	0.7047E-04
12	129600.0	4.7428	0.1469E-04	158.0031	0.6056E-04
13	172800.0	8.4769	0.3482E-04	161.7687	0.6976E-04
14	86400.0	3.5248	0.1175E-04	155.6485	0.7361E-04
15	129600.0	5.1084	-0.4343E-06	159.0629	0.6081E-04
16	172800.0	5.9240	0.1422E-04	163.7965	0.7615E-04
17	216000.0	6.7804	0.1130E-04	164.9938	0.7198E-04
18	43200.0	0.3364	0.0000E+00	154.4359	0.0000E+00
19	172800.0	0.5223	0.0000E+00	149.6579	0.0000E+00
20	0.0	7.5385	0.4305E-04	159.0714	-0.6106E-05
21	172800.0	9.7906	0.2576E-04	157.7525	0.1597E-04
22	0.0	6.4738	0.2285E-04	158.6602	-0.4225E-05
23	43200.0	7.2620	0.2001E-04	159.0805	-0.3449E-04
24	86400.0	8.2195	0.3025E-04	159.2360	0.6079E-05
25	172800.0	11.4950	0.1706E-04	158.0335	-0.8865E-05
26	86400.0	4.7811	0.3463E-04	150.2000	-0.2200E-04
27	216000.0	13.4423	0.7618E-04	151.3719	-0.2750E-04
28	43200.0	6.6095	-0.5367E-04	155.0578	0.1522E-04
29	129600.0	-0.9081	-0.7107E-04	157.1149	0.2712E-04
30	216000.0	-6.2626	-0.9251E-04	159.4644	0.2897E-04

Table 2-6. Sensor characteristics for simulated MSI data bases.

Run Number	Data Base	Sensor Type	Measurement Standard Deviation*				Detection Probability
			Latitude (deg^2)	Longitude (deg^2)	Latitude Rate $(deg/hr)^2$	Longitude Rate $(deg/hr)^2$	
1	ORT1A	1	.1	.1	.12E-5	.12E-5	.5
		2	.1	.1	Not measured	Not measured	.5
2	ORT1B	1	1.	1.	.12E-4	.12E-4	.5
		2	1.	1.	Not measured	Not measured	.5

*Measurement noise is modeled as a zero-mean uncorrelated Gaussian process with the indicated standard deviation.

The individual track files are contained in both synthetic data bases. As Table 2-6 indicates, the only difference in data bases ORT1A and ORT1B is the measurement accuracy of the sensors. The simulated sensor accuracy in data base ORT1A is ten times greater than that in ORT1B.

The test cases involving simulated data were run on a DEC 10 computer.

2.2 Problem Structure for Simulated MSI Scenarios

The basic structure of the k-track correlator is that of an integer program that operates on a set of potential correlations to select the most accurate picture possible of the surveillance area.

The set of potential correlations for Run #1 selected by the backtracking Kalman filter process described in Section 1.2.1 is shown in Figures 2-2 through 2-12. Each figure indicates the track file data points involved in that particular correlation. Although the parameter file setup allowed as many as ten tracks to be correlated simultaneously, the various tests employed during the backtracking process selected only 2- and 3-track correlations.

Three basic types of errors can occur in the process of forming potential k-track correlations:

Type 1 Error: the tracks included in the correlation are correct (in that they correspond to the same ship), but not all tracks from the same ship were included.

Type 2 Error: the correlation includes tracks from more than one ship.

Type 3 Error: failure to detect a true correlation.

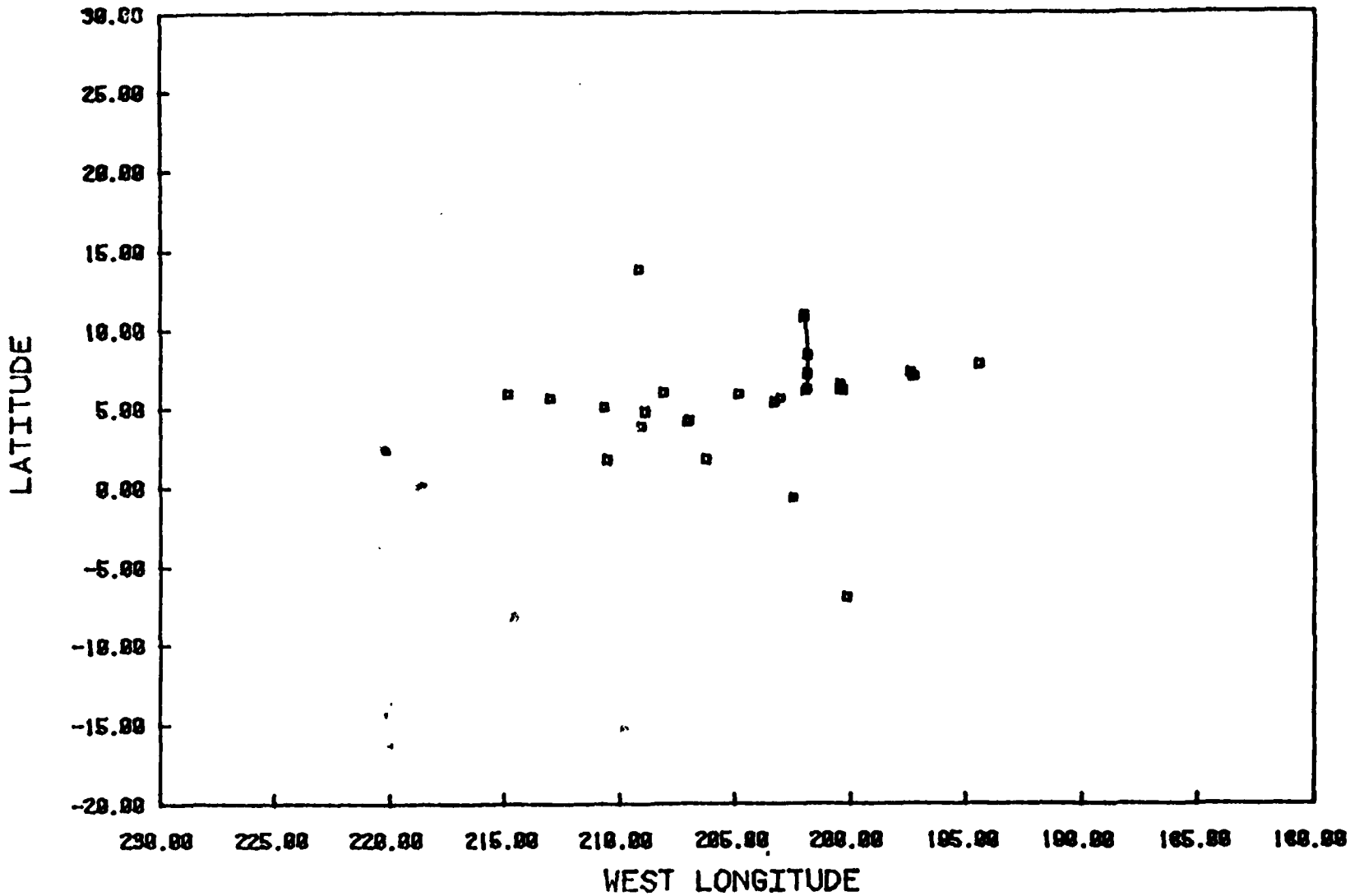


Figure 2-2. Potential correlation #1 (ORT1A/PAR1A).

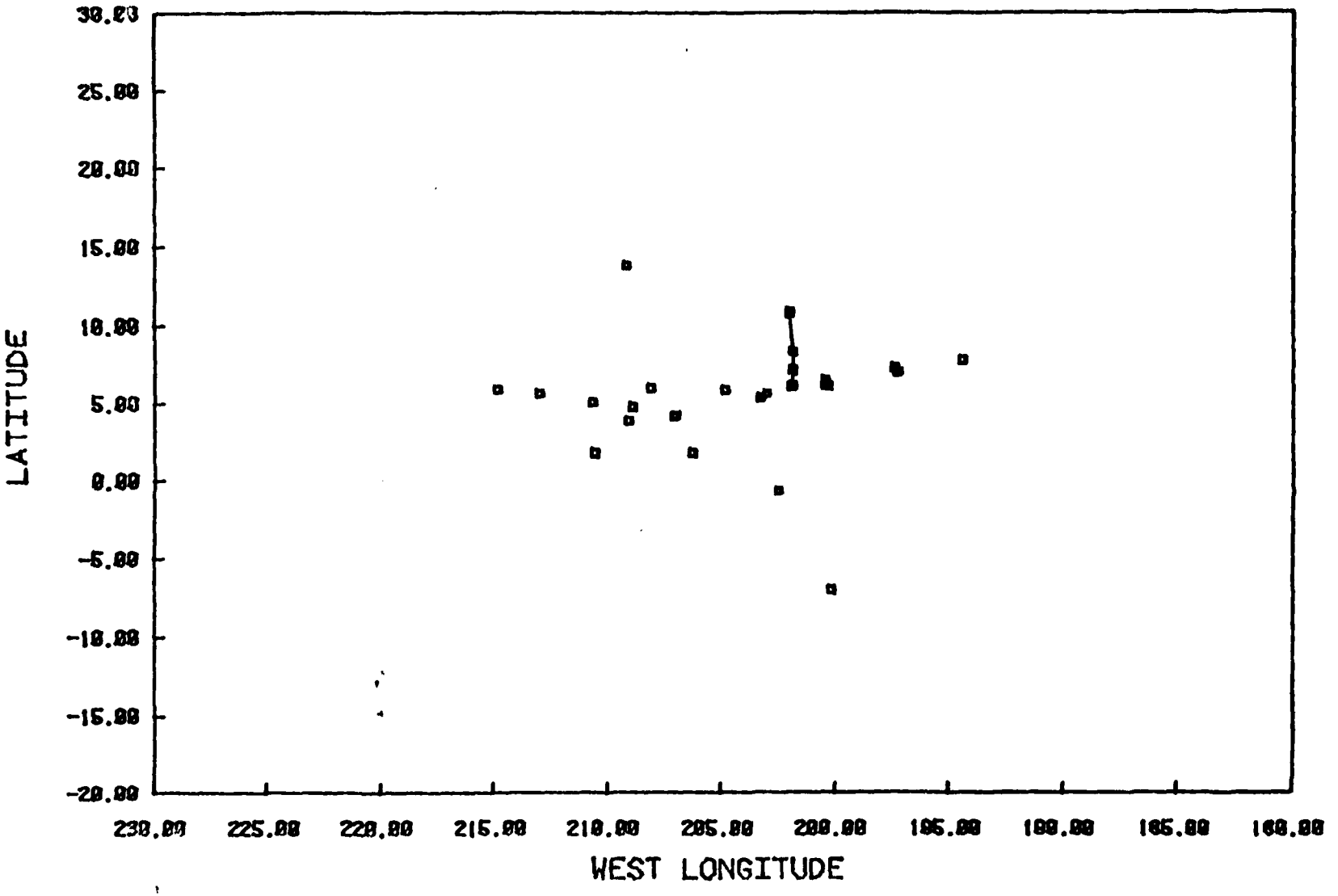


Figure 2-3. Potential correlation #2 (ORT1A/PAR1A).

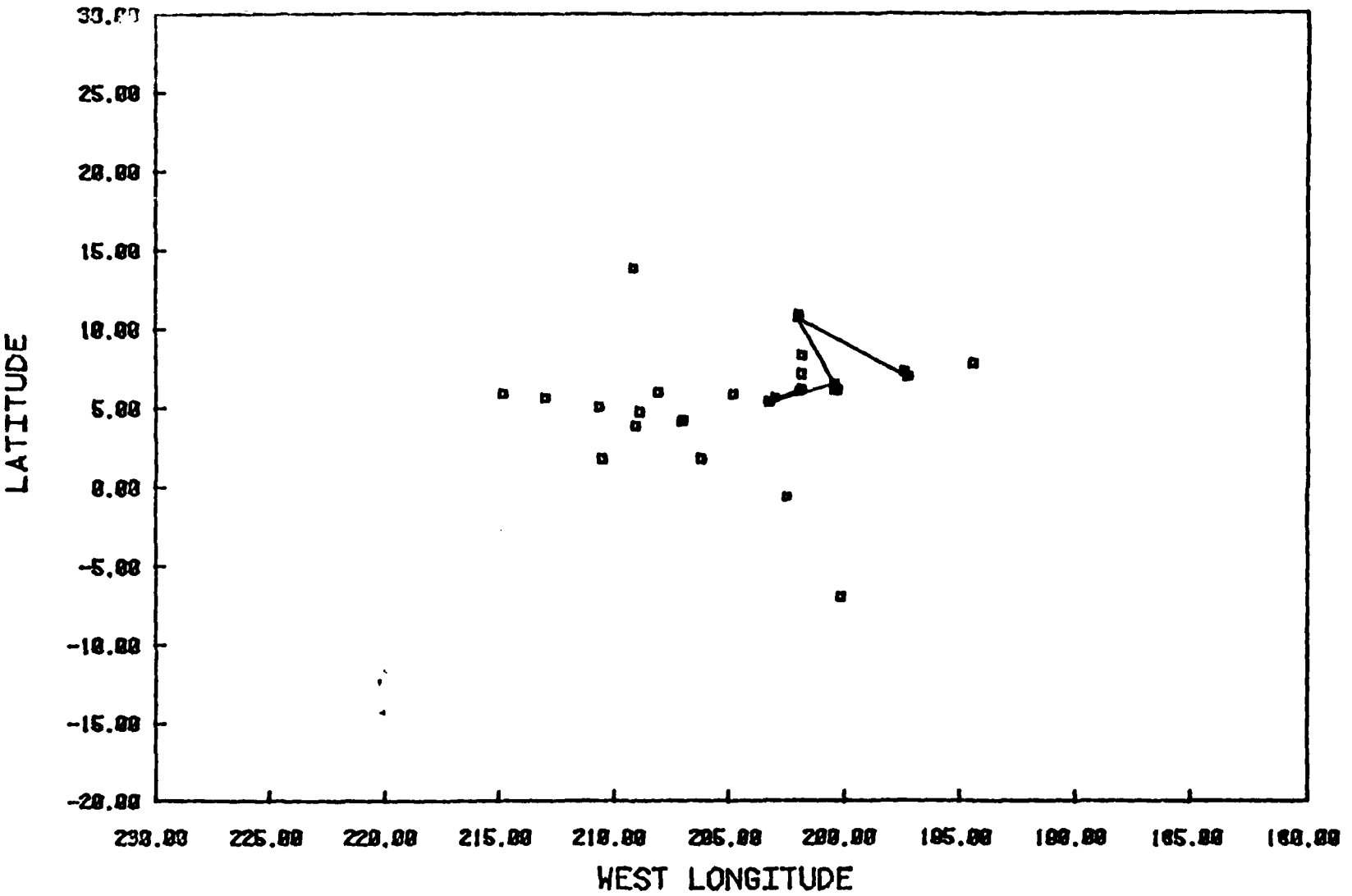


Figure 2-4. Potential correlation #3 (ORT1A/PAR1A).

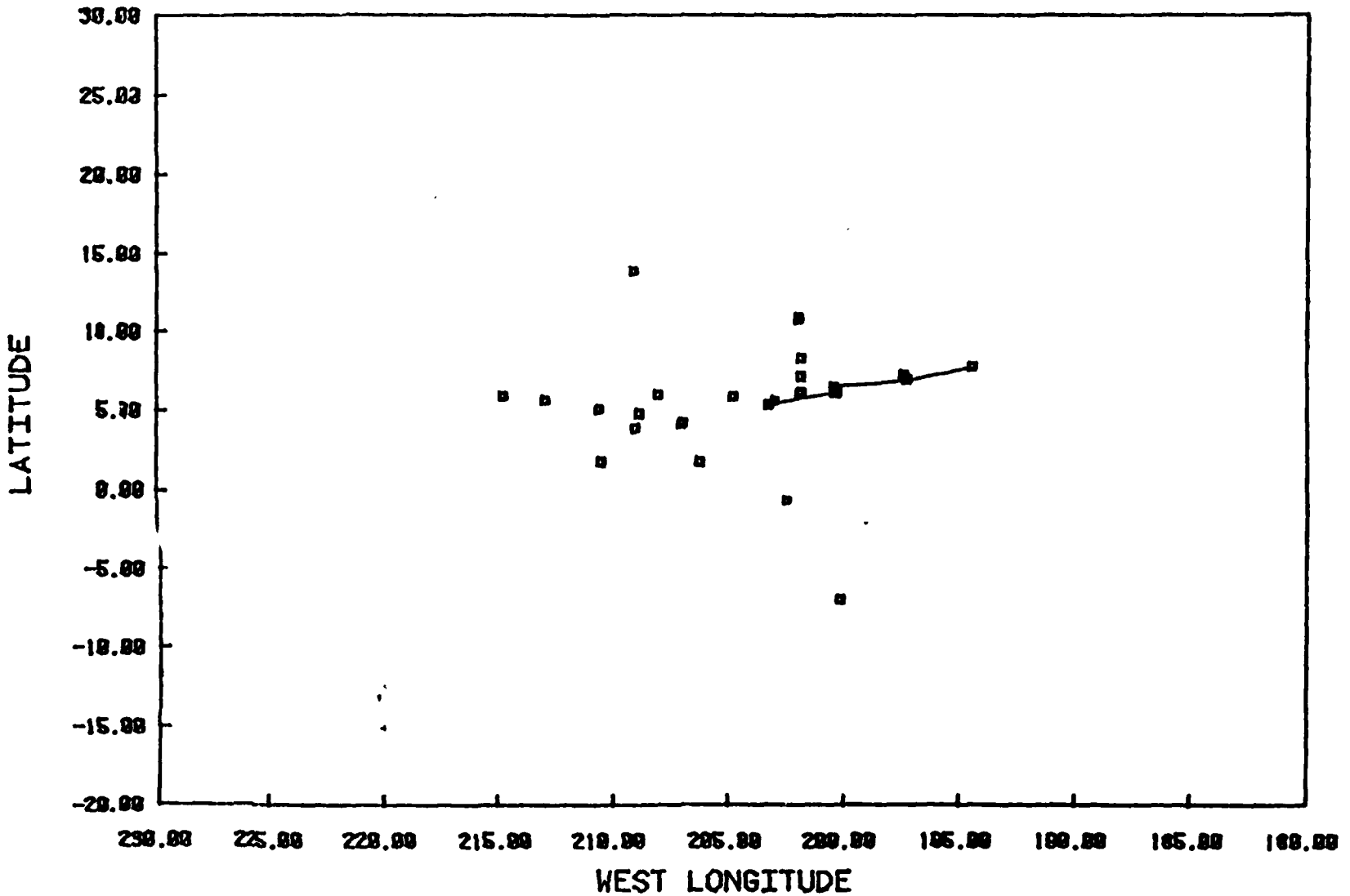


Figure 2-5. Potential correlation #4 (ORT1A/PAR1A).

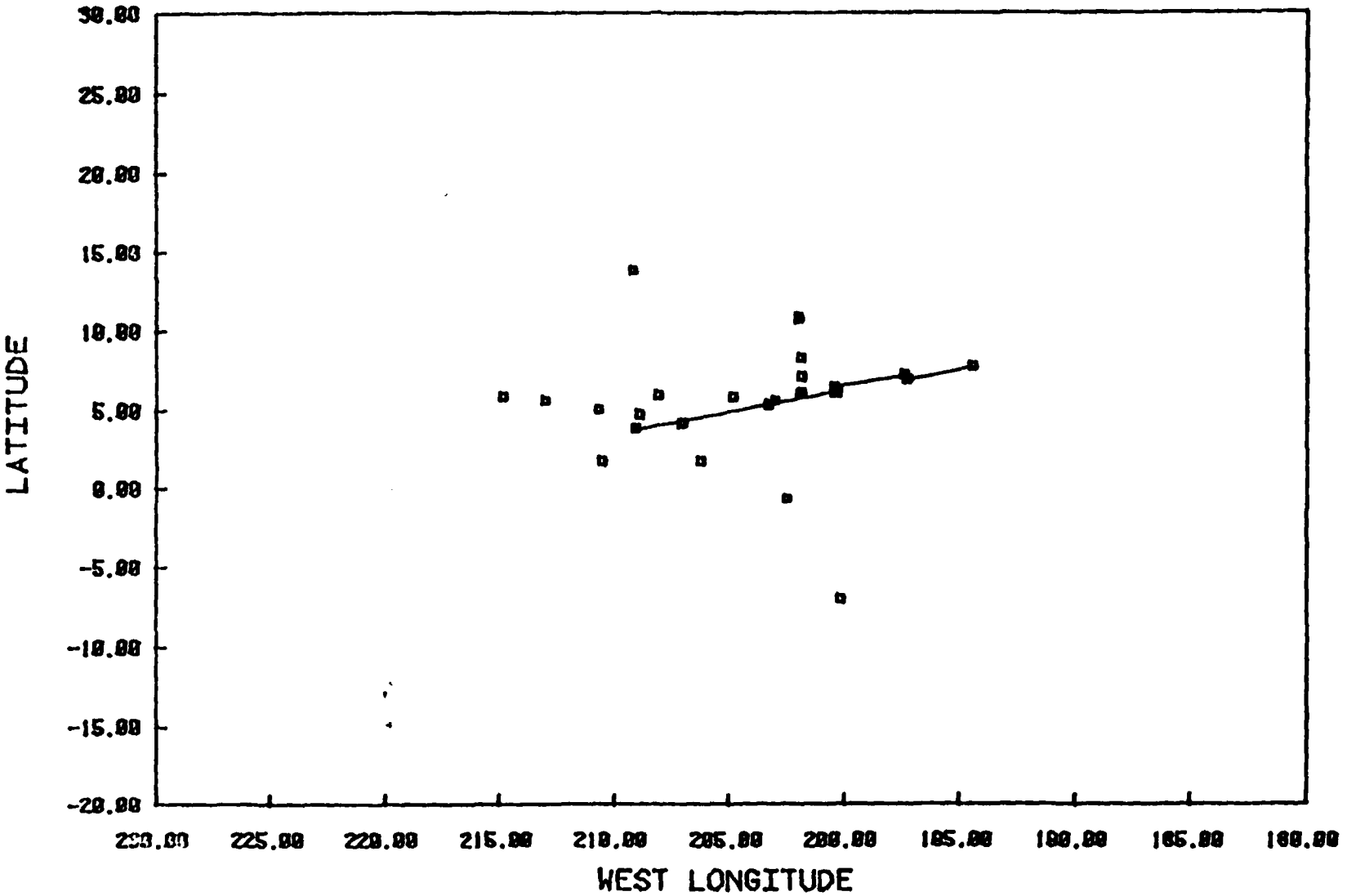


Figure 2-6. Potential correlation #5 (ORT1A/PAR1A).

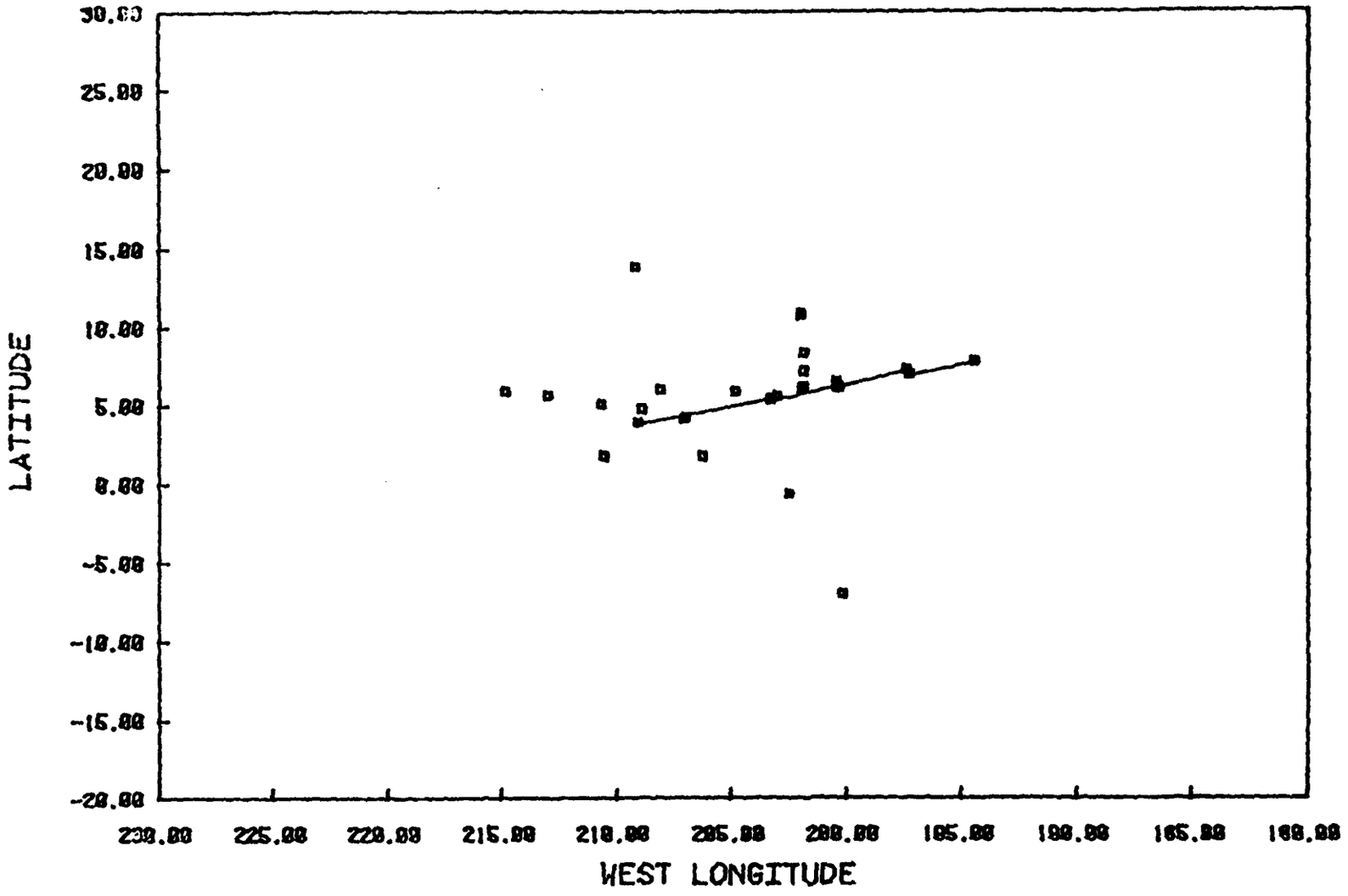


Figure 2-7. Potential correlation #6 (ORT1A/PAR1A).

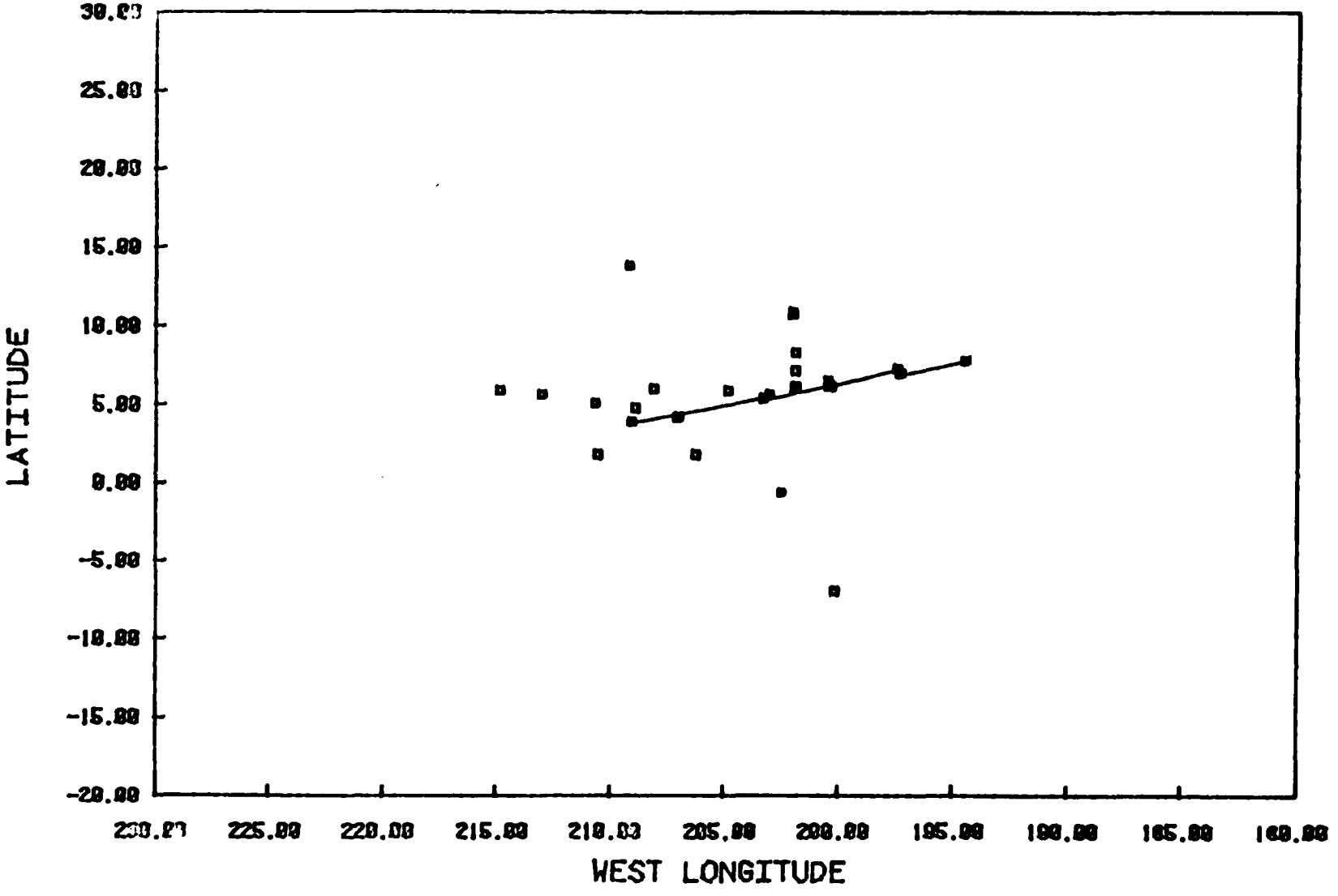


Figure 2-8. Potential correlation #7 (ORT1A/PAR1A).

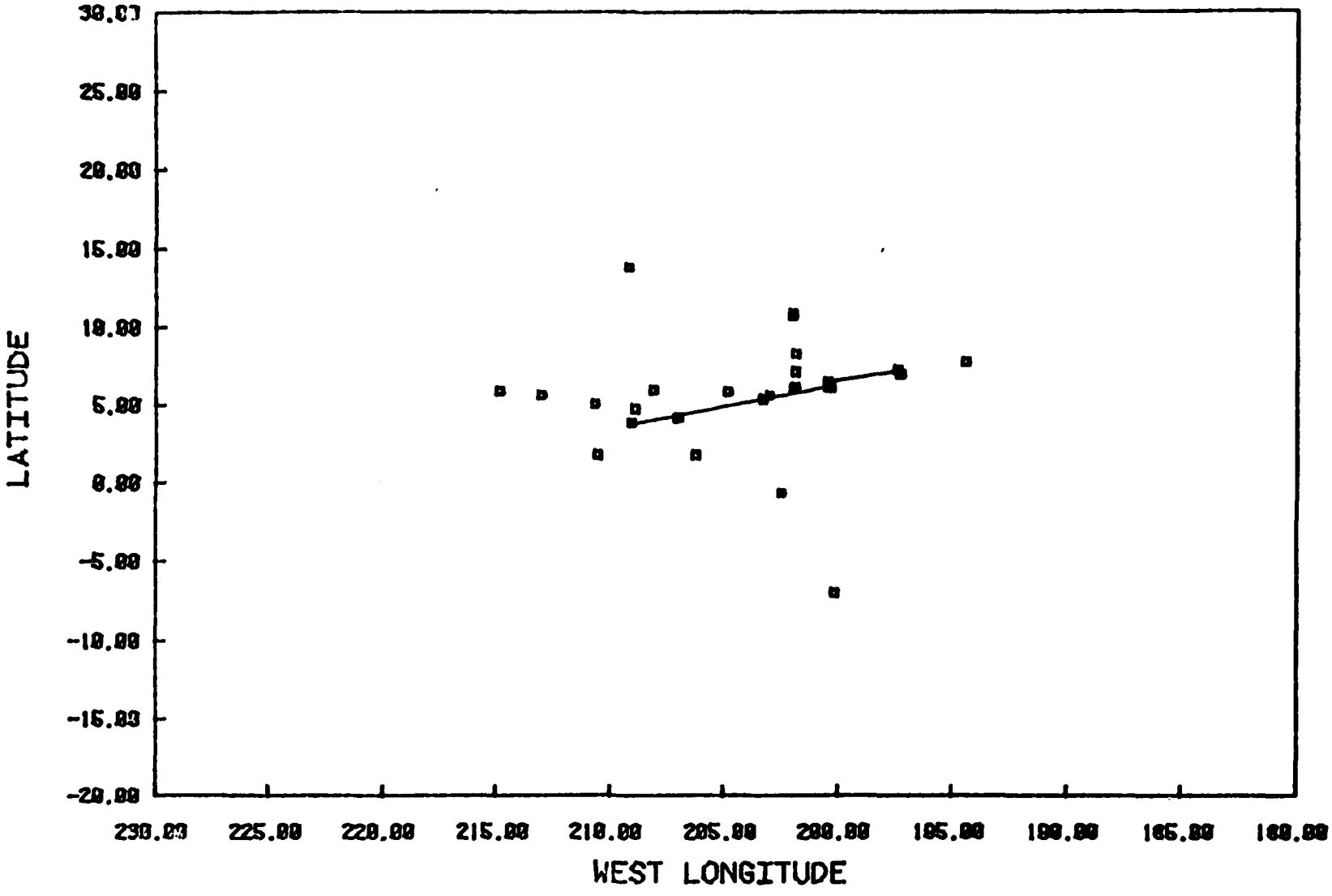


Figure 2-9. Potential correlation #8 (ORT1A/PAR1A).

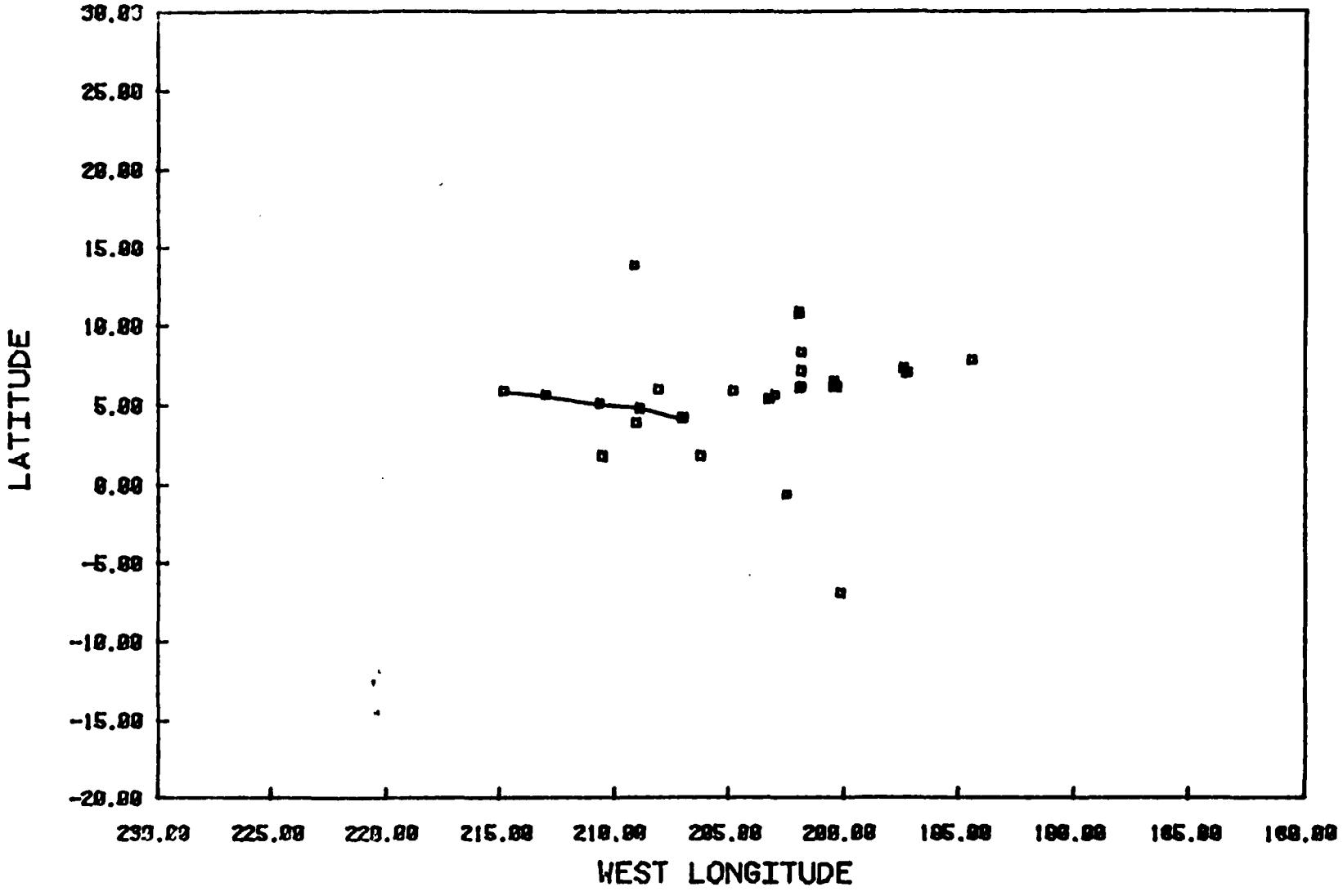


Figure 2-10. Potential correlation #9 (ORT1A/PAR1A).

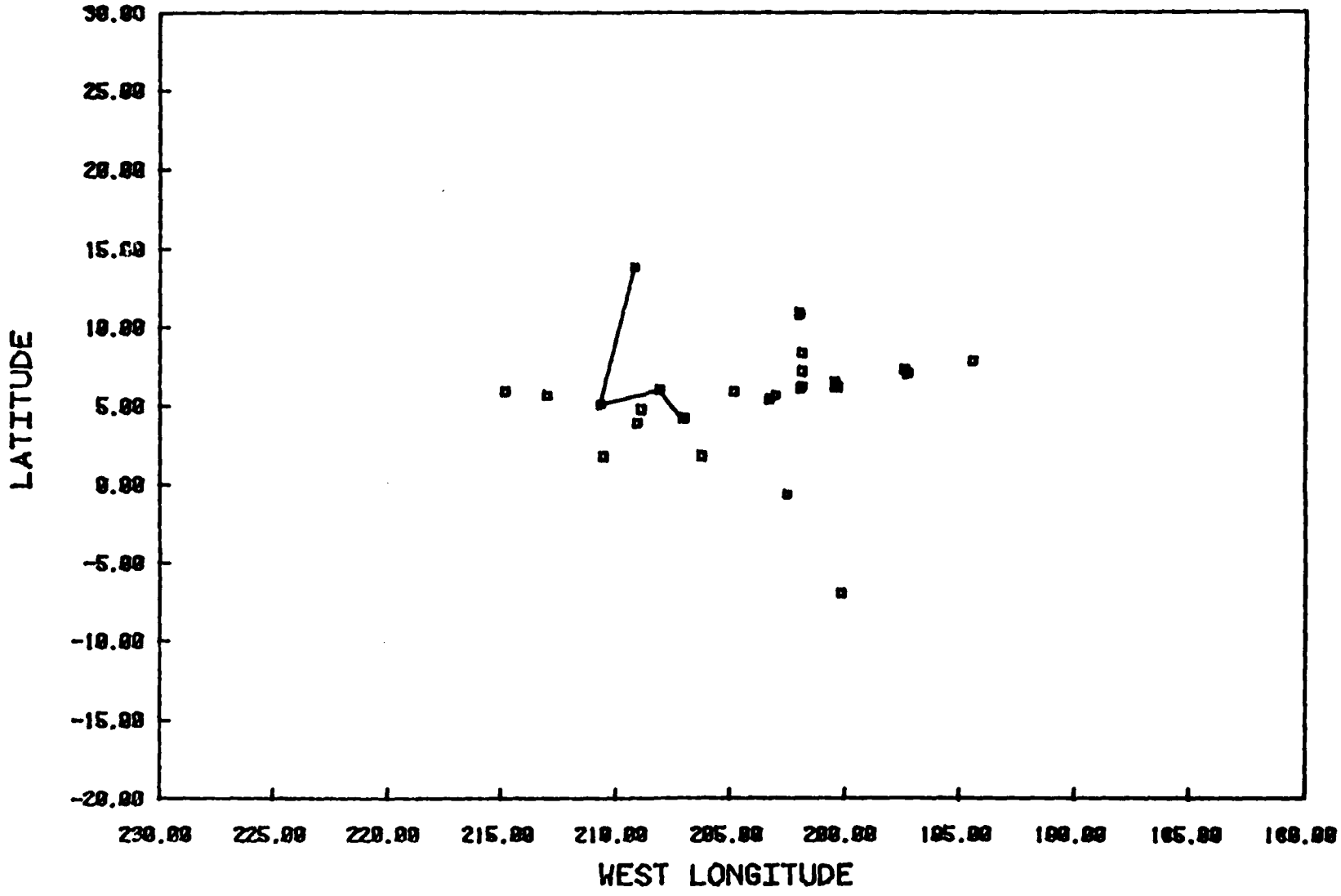


Figure 2-11. Potential correlation #10 (ORT1A/PAR1A).

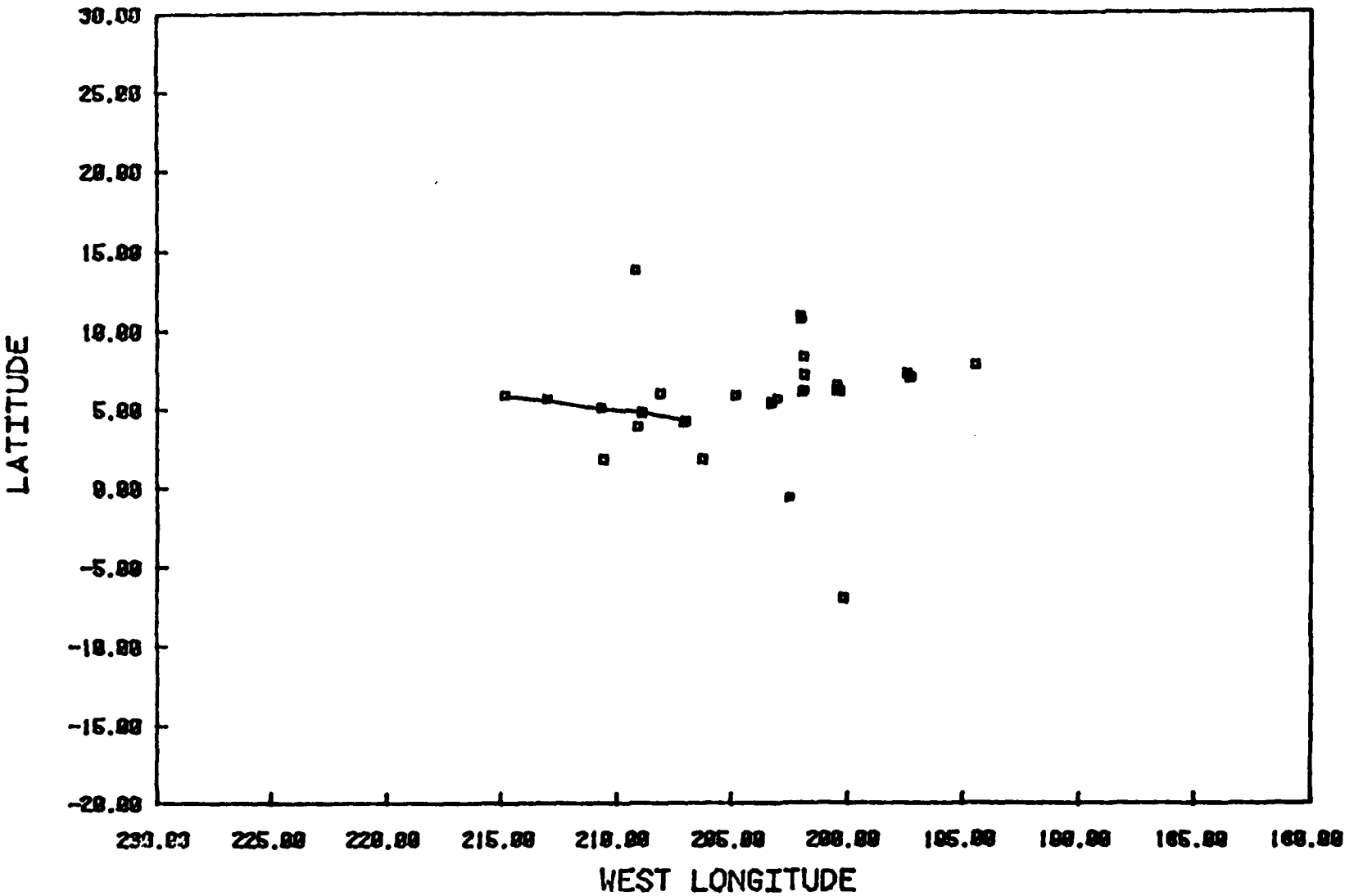


Figure 2-12. Potential correlation #11 (ORT1A/PAR1A).

Table 2-7 below summarizes the results of the backtracking Kalman filter on a correlation-by-correlation basis. Note that both Type 1 and Type 2 errors are mixed in with the set of correct correlations. No Type 3 errors were noted in either run.

Table 2-7. Summary of potential correlations constructed during Run #1.

Correl Number	Time	Data Point Number	Comments
1	.00000E+00 .00000E+00 .12000E+02 .24000E+02 .48000E+02 .48000E+02	22 20 23 24 25 21	Correct. Correlates Tracks #7 and #8.
2	.00000E+00 .00000E+00 .12000E+02 .24000E+02 .48000E+02 .48000E+02	20 22 23 24 21 25	Same as above. Note that data points were reordered due to coincidental time of report.
3	.00000E+00 .24000E+02 .36000E+02 .48000E+02 .48000E+02	20 7 8 21 9	Incorrect (Type 2 error).
4	.24000E+02 .24000E+02 .36000E+02 .36000E+02 .48000E+02 .48000E+02 .60000E+02	14 7 15 8 16 9 17	Incorrect (Type 1 error). Correlates Tracks #3 and #5, but leaves out Track #4.
5	.00000E+00 .24000E+02 .24000E+02 .24000E+02 .36000E+02 .36000E+02 .48000E+02 .48000E+02 .60000E+02	19 14 14 7 12 15 8 9 17	Correct. Correlates Tracks #3, #4, and #5.

Table 2-7. (Continued)

Correl Number	Time	Data Point Number	Comments
6	.00000E+00 .24000E+02 .24000E+02 .36000E+02 .36000E+02 .48000E+02 .48000E+02 .60000E+02	10 11 14 12 15 13 16 17	Incorrect (Type 1 error). Correlates Tracks #4 and #5, but leaves out Track #3.
7	.00000E+00 .24000E+02 .24000E+02 .24000E+02 .36000E+02 .36000E+02 .36000E+02 .48000E+02 .48000E+02 .48000E+02 .60000E+02	10 11 7 14 12 8 15 13 9 16 17	Same as Correlation 5. Note that data points were reordered due to coincidental time of report.
8	.00000E+00 .24000E+02 .24000E+02 .36000E+02 .36000E+02 .48000E+02 .48000E+02	10 11 7 12 8 13 9	Incorrect (Type 1 error). Correlates Tracks #3 and #4, but leaves out Track #5.
9	.12000E+02 .12000E+02 .24000E+02 .36000E+02 .48000E+02 .60000E+02	5 6 7 8 9 10	Correct. Correlates Tracks #1 and #2.
10	.12000E+02 .24000E+02 .36000E+02 .60000E+02	6 7 8 10	Incorrect (Type 2 error).
11	.12000E+02 .12000E+02 .24000E+02 .36000E+02 .48000E+02 .60000E+02	6 7 8 9 10	Same as Correlation 9. Note that data points were reordered due to coincidental time of report.

Note that in Table 2-7 each individual correlation consists of combinations of complete track files. For example, correlation 10 consists of data points {1, 26, 2, 27}, and we don't see correlations such as {1, 26, 2}. This particular problem structure is the defining characteristic of the track-to-track correlation problem. It arises because it is assumed that complete tracks (e.g., {1, 2} and {26, 27}) are constructed properly at the sensor level. (By contrast, in a sensor-level multitarget tracking problem, the key problem is to decide whether or not data points 1 and 2 are the same ship [1].)

The structure of the 0-1 integer program for k-track correlation can be illustrated by converting the potential correlations for Run #1 into the form of Equations (11) through (13), repeated here for convenience:

$$\max \; d^t \nu$$

subject to

$$B\nu \leq \mathbf{1}$$

$$\nu \text{ binary} .$$

For Run #1, this problem has the following form:

Potential Correlation Number ⟶

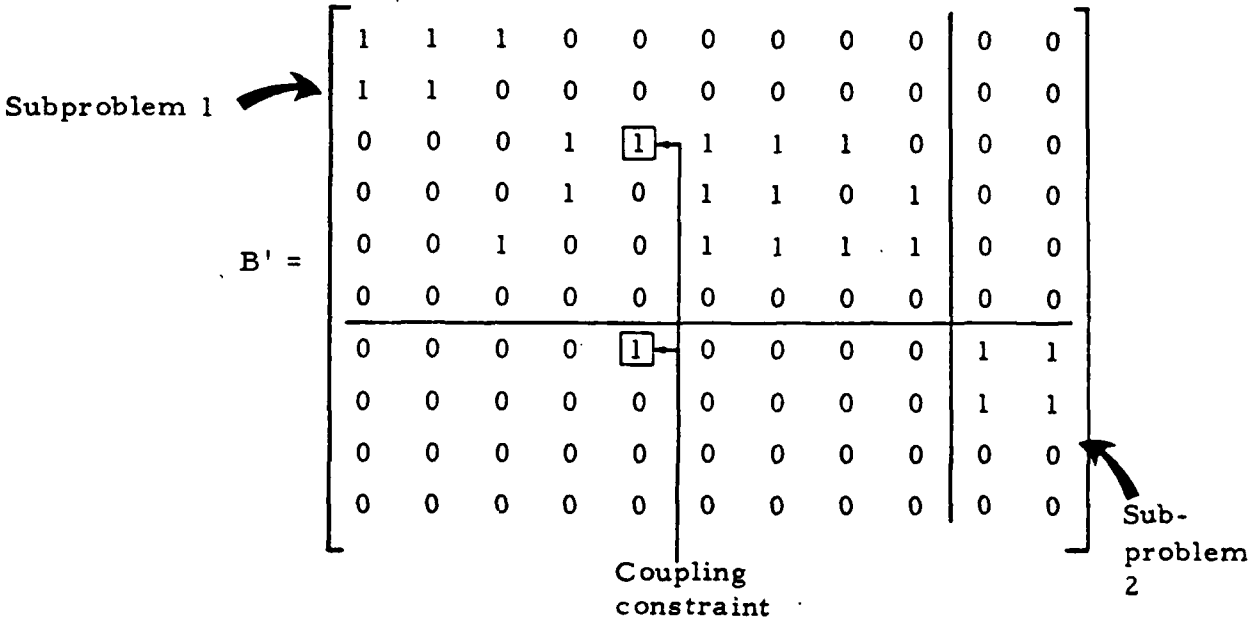
$$d = \begin{bmatrix} 130. \\ 130. \\ -1217. \\ 96. \\ 187. \\ 174. \\ 187. \\ 95. \\ 126. \\ -678. \\ 126. \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Track File Numbers ⟶

The vector d has elements which correspond to the log-likelihood function (Equation (8)) for each potential correlation. Thus the first element of d is the numerical value of $\ell n\, p(\omega)$ evaluated at the last data point in Correlation 1.

Note that B is a 10x11 matrix (number of track files x number of potential correlations). B is made up of columns of zeros and ones, each column representing one particular correlation. For example, Column 1 in B represents Correlation 1, which consists of Tracks #7 and #8 (the 7th and 8th elements of the column are set to 1).

Closer examination of B reveals that the problem can be decomposed into a set of independent subproblems. By appropriately permuting columns of B, we have the equivalent matrix



1	1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
0	0	0	1	[1]	1	1	1	0	0	0
0	0	0	1	0	1	1	0	1	0	0
0	0	1	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	[1]	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Thus B' is a decomposition of the original intersensor correlation problem into two subproblems, with one coupling constraint.

The structure of the problem is further reduced by column elimination. Let b^i be the i-th column of B. Then the j-th column of B can be eliminated if

$$b^i = b^j$$

and

$$d_i \geq d_j ,$$

where d_i is the i-th element of the cost vector d. Furthermore, the k-th column can be eliminated if $d_k \leq 0$. Finally, any row of zeros can be eliminated. Carrying out this process for B', we arrive at an integer program with maximum decoupling and minimum size (note that the coupling constraint was eliminated, and that Subproblem 1 was further decomposed):

Subproblem 1a

$$d'' = \begin{bmatrix} 126. \\ 95. \\ 187. \\ 96. \\ 174. \\ 130. \end{bmatrix}, \quad B'' = \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Subproblem 1b

Subproblem 2

Application of these methods markedly decreases run time, which depends exponentially upon the size of B.

2.3 Algorithm Characteristics

The truth models and results of integer program processing for Runs #1 and 2 are shown graphically in Figures 2-13 through 2-16. As the figures indicate, in Run #1 the k-track correlation algorithm performed correctly. In Run #2, with sensors one order of magnitude less accurate (ratio of standard deviation of noise), one Type 1 error was made.

Note that the integer program is constrained to select from among the elements of potential correlation set D during the process of forming a complete picture of the surveillance area. Since Type 1, Type 2, and Type 3 errors can occur during the formation of the set D, the integer program is subject to errors of the same type.

The total run time requirement of the k-track correlation algorithm depends upon the run times of the two individual segments of the code: (1) the construction of potential correlations, and (2) the integer program. Of the two segments, the integer program is the most sensitive to program size. As Figure 2-17 indicates, the results for ten test cases indicate a reasonable growth in the back-tracking Kalman filter run time as the problem size increases. As Figure 2-18 indicates, the integer program exhibits a sustained exponential increase in run time as problem size increases. The underlying problem structure is such that the decoupling procedure mentioned in Section 2.2 above is critical for MSI problems where extremely large numbers of potential correlations are found.

In several cases, a suboptimal solution was returned by the integer program due to run time constraints placed on the code. In those cases where the maximum time limit resulted in a possibly sub-optimal solution, good accuracy was obtained. Thus it appears that good suboptimal solutions to the integer program can be obtained within a reasonable period of time for the types of data bases discussed in this paper.

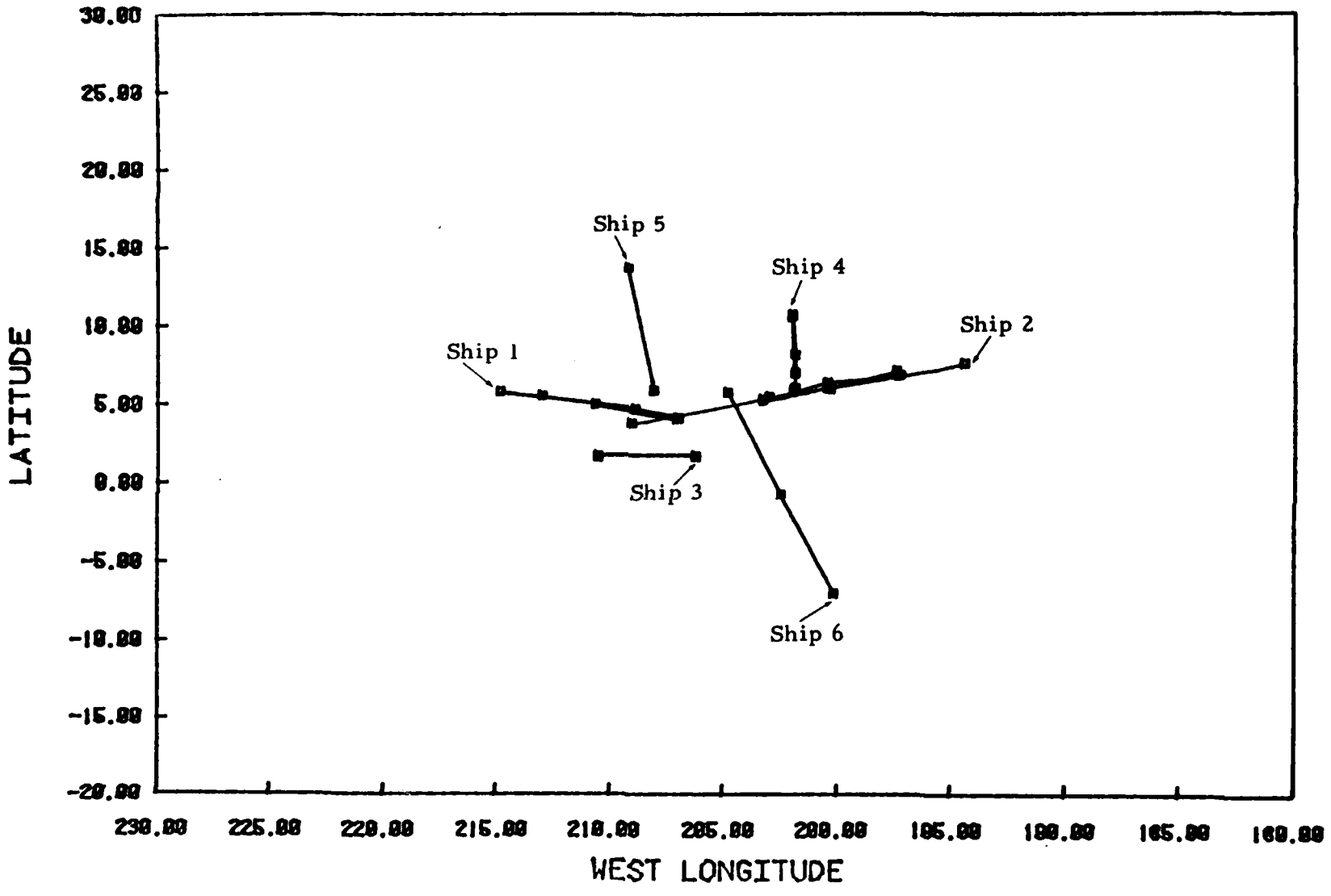


Figure 2-13. Actual tracks contained in synthetic data base ORT1A.

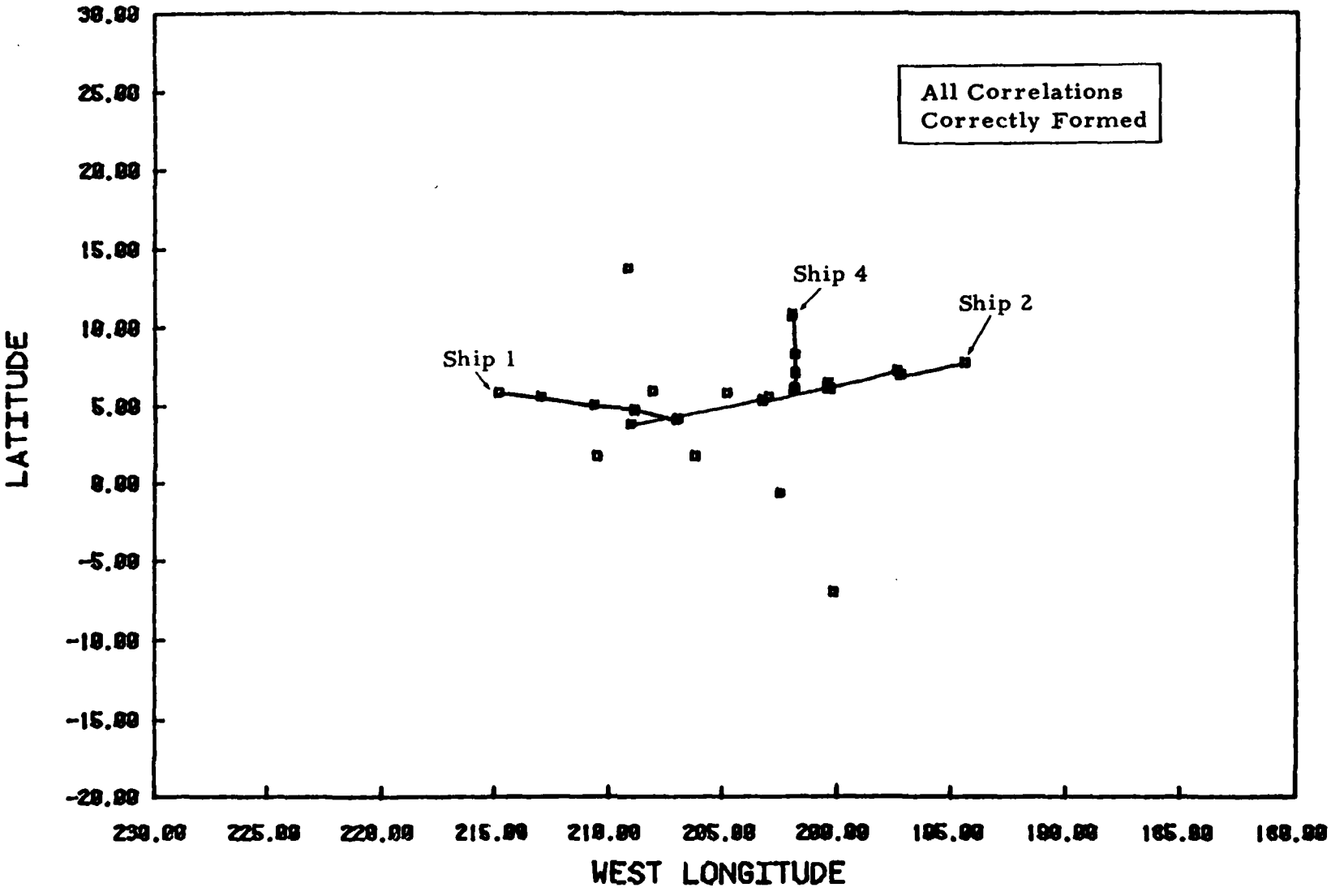


Figure 2-14. Correlated tracks produced by integer program (ORT1A/PAR1A).

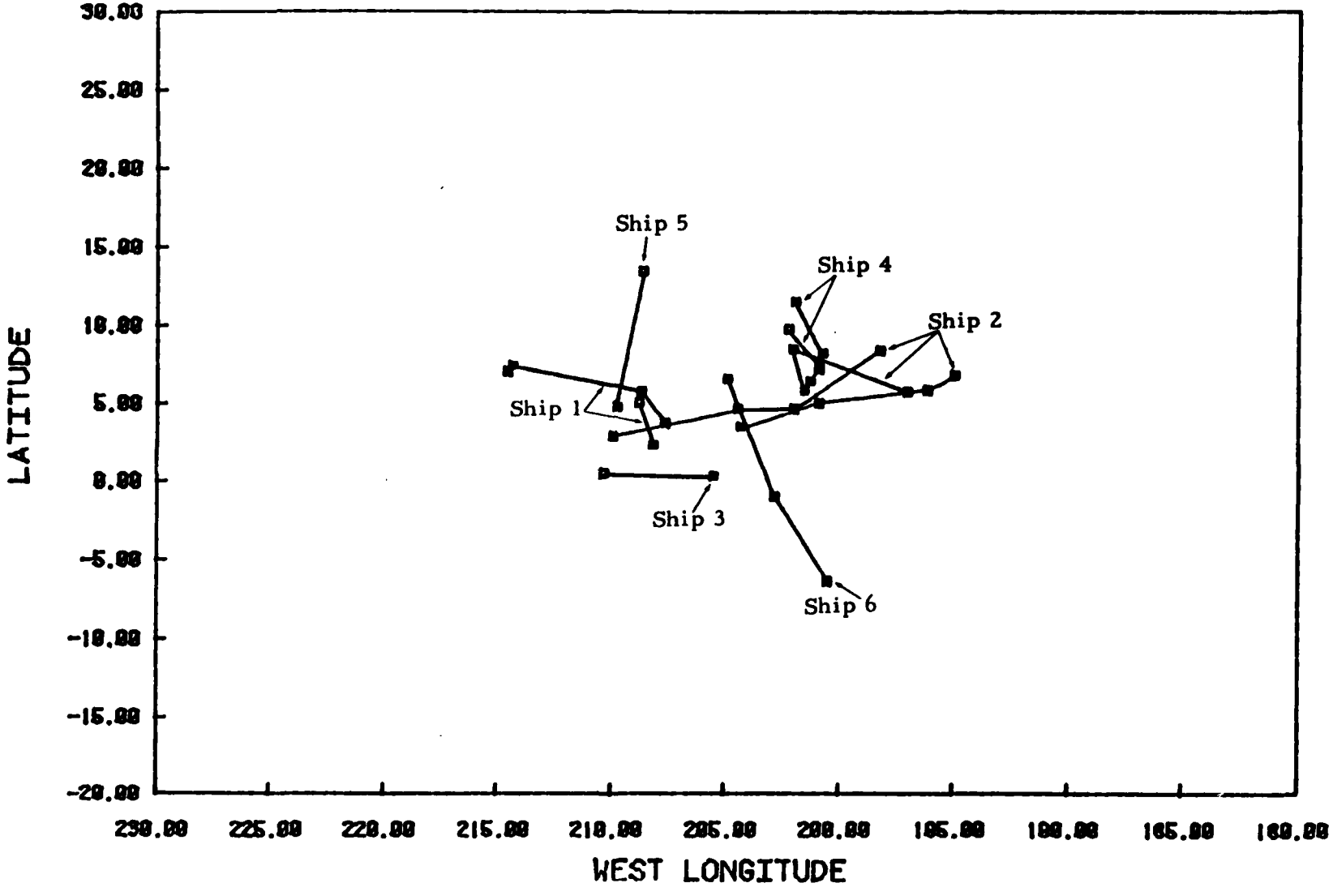


Figure 2-15. Actual tracks contained in synthetic data base ORT1B.

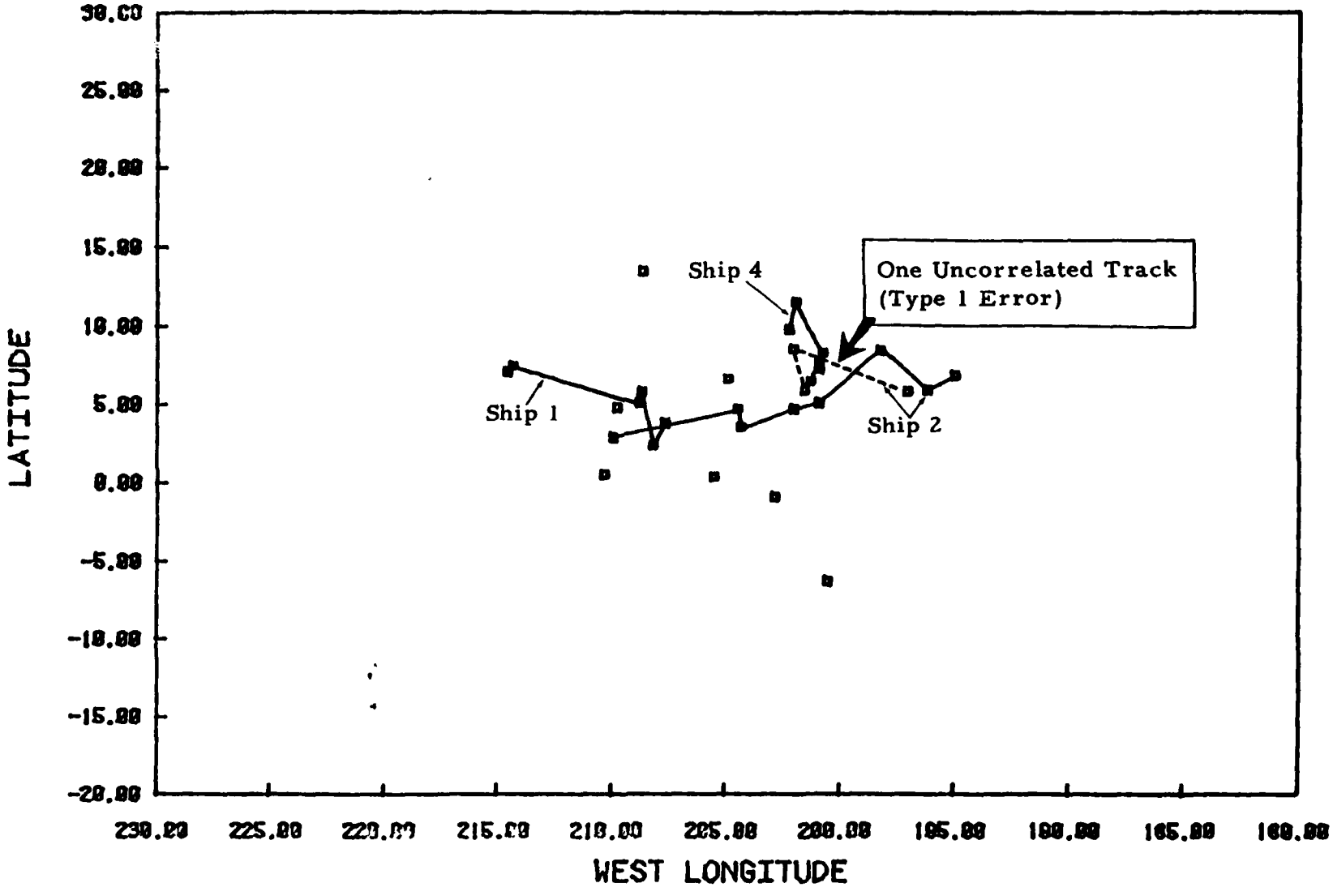


Figure 2-16. Correlated tracks produced by integer program (ORTIB/PARIB).

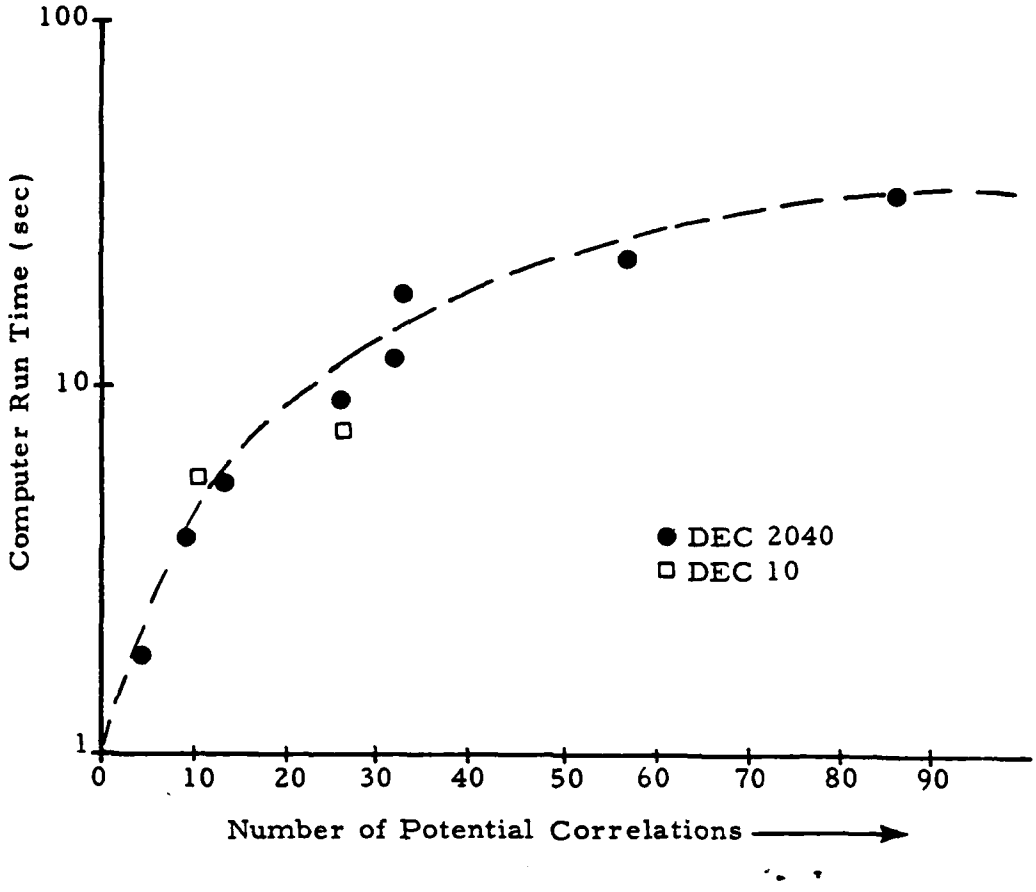


Figure 2-17. Run times for construction of set of potential correlations.

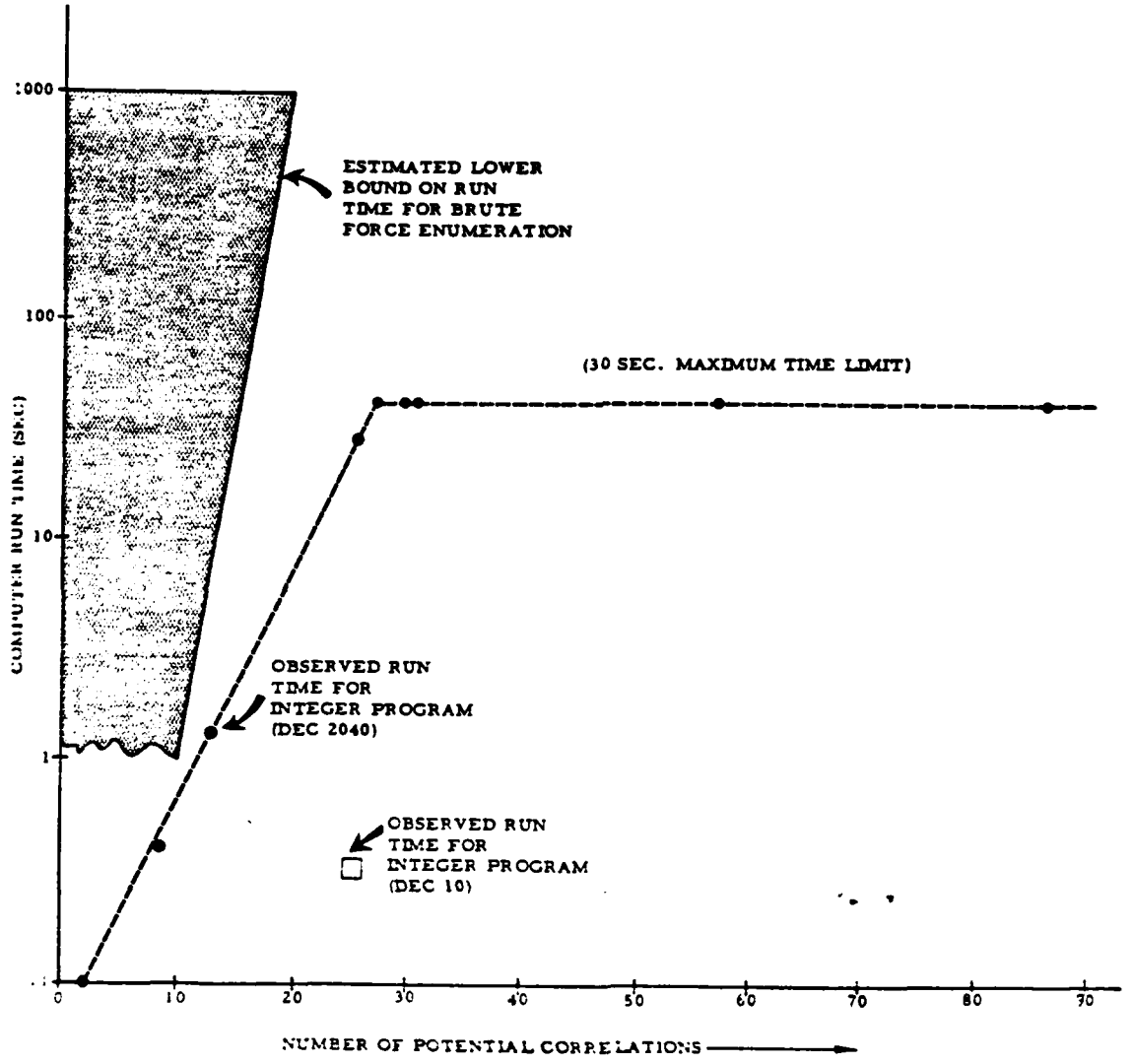


Figure 2-18. Run times for solution of Bayesian decision problem.

3.0 CONCLUSIONS

This paper summarizes a study which consisted of essentially two parts:

- initial development of a sophisticated intersensor (track-to-track) correlation algorithm
- preliminary analysis of the algorithm's characteristics when applied to synthetic data.

The primary finding of the paper is that an accurate picture of a large ocean surveillance area can be constructed automatically by the k-track correlation algorithm. This finding is substantiated by the analysis of synthetic data discussed in Section 2.

The structure of the algorithm is that of a Bayesian decision process, which by its very nature may produce some errors in the analysis of any particular data base. As noted above, these errors were minimal for the analysis performed herein. There is a critical need for parameter tuning against truth models before the algorithm can be trusted in an operational situation. Thus it is clear that substantial work remains before the promising results obtained during this short preliminary study can be broadly applied.

In the area of track-to-track correlation algorithm development, there remains a number of important issues yet to be resolved. A few of these are as follows:

- How large a problem can the integer program handle in real time after the decoupling procedure mentioned in Section 2.2 is implemented?
- Using the k-track correlation algorithm as a benchmark, how accurate are simpler suboptimal correlation algorithms (such as the pairwise algorithm mentioned in Section 1.2)?

- What is the impact on algorithm accuracy of Kalman filter mismatch (i.e., a mismatch between actual and assumed sensor accuracy)?
- What is the feasible operating regime for track-to-track correlation algorithms in terms of ship density?
- How sensitive is algorithm accuracy to intersensor bias? How accurately can intersensor alignment be carried out? Can we "bootstrap" the alignment as track-to-track correlation is carried out?
- What is the proper balance between real-time track-to-track correlation accuracy and the load on surveillance network communication links (how much data is enough for a given surveillance area)?

This partial list of important MSI issues yet to be resolved can of course be substantially expanded. Hopefully, the results of this short preliminary study of one specific correlation algorithm will resolve some of the complex issues surrounding automatic, accurate, real-time ocean surveillance.

REFERENCE

[1] C. L. Morefield, "Application of 0-1 Integer Programming to Multitarget Tracking Problems," IEEE Trans. Auto. Contr., Vol. AC-22, June, 1977, pp. 302-312.